Enterprise Common Components

Installation and Customization Guide
MVS Version

Release 1.6

COMPUWARE.

Please direct questions about Enterprise Common Components
or comments on this document to:

Enterprise Common Components Technical Support
Compuware Corporation
One Campus Martius
Detroit, MI 48226-5099
1-800-538-7822

Outside the USA and Canada, please contact
your local Compuware office or agent.

Doc. CWCYIX1F
January 5, 2004.

Contents

Figures

Tables

Summary of Changes

Enterprise Common Components Release 1.6

Enterprise Common Components (ECC) 1.6 incorporates Compuware Shared Services (CSS) upgrade from Release 7.9 to 8.0 and the inclusion of Host Communication Interface (HCI) release 2.5 installation instructions.

Compuware Shared Services 8.0

CSS Release 8.0 offers the following new features and enhancements:

- Additional source support for Enterprise PL/I for Abend-AID. External and controlled storage areas are now supported.
- Source support and mismatch capabilities for VisualAge and Enterprise PL/I within Abend-AID for CICS is now supported.
- Additional cursor-selectable fields have been added within the Abend-AID report between the COBOL file-definition ('FD') program storage and the file sections.
- Additional cursor-selectable panel selection capability has been incorporated within the Abend-AID report for all of the supported languages. Highlighted fields have been added in the Call Trace Summary, Program Listing, and Program Storage sections.
- Enhanced support for COBOL BLF cells. FD information is not always available during the time an Abend-AID report is captured. A more informative message is displayed when viewing program storage when a specified file is not open.
- Users will now be able to adjust the display of vertical hex storage to be relative to zero or one. These commands are available in the File section of the Abend-AID report and additionally in the Sort Information section for COBOL programs.
- The Output Selection Menu of an Abend-AID report will now signify which sections are available by highlighting (reverse-video) those options. If a section that is not available for that particular report is selected, a descriptive message will be displayed stating why that section is not available.
- The time used during the CreateMem process during Abend-AID for CICS dump capture time has been reduced thus freeing up system resources more quickly.
- An improved interface for XPEDITER/CICS DDIO processing has been provided to enhance overall Compuware Shared Services product performance and resource management. This includes dynamically allocating/deallocating source listing files.
- Source shared directory functionality for all supported languages can now be utilized by XPEDITER/TSO release 7.3 and XPEDITER/CICS release 7.6.

  Note: XPEDITER/TSO releases 7.0, 7.1, and 7.2 allow full Shared Directory support for C language and PL/I LONGNAME support only. XPEDITER/CICS releases 7.4 and 7.5 as well as XPEDITER/TSO 7.0, 7.1, and 7.2 support the use of source databases for all languages as standalone DDIO files only.

- A DDIO calculator utility is available in this release to compute the optimal DDIO formatting parameters based on various input in the form of a DIRX report or the actual file.
- Abend-AID/XLS, or source support, will now be referred to as simply Abend-AID.

Support for Earlier Releases

- The Viewing Facility has removed support for the CICS Abend-AID product. CICS Abend-AID was the pre-cursor product to CICS Abend-AID/FX, which was renamed Abend-AID for CICS in this release. CICS Abend-AID/FX implemented a Viewing Server for online dump analysis, so support in the Compuware Viewing Facility was no longer needed.

Host Communications Interface 2.5

HCI Release 2.5 offers the following new features and enhancements:

- **UNIFACE Version 8 Support.**

  Uniface version 8 is fully supported.

- **New CWETPS VERB.**

  The Extract_TP_Status function (CWETPS) can be used to determine if a TP has registered with the ECC.

- **Fully Supported IP Name.**

  In the previous releases, the IP name could not be specified with the TP name in the HCICNSIT definition. This has been changed to permit both to be used at the same time. This change is particularly important to TCP protocols.

- **More Concurrent Conversations Supported.**

  The HCICNGCA MAXCCBS value has a new maximum - 4096. This permits very large sites to drive a larger number of conversations.

- The Host Communications Interface (HCI) Release 2.5 component will now be distributed as a separate component exclusively on the ECC installation tape.

Base Services 5.2

The Base Services 5.2 component will now be distributed as a separate component exclusively on the ECC installation tape.

Enterprise Common Components Release 1.5

Enterprise Common Components 1.5 incorporates the Compuware Shared Services upgrade from Release 7.8 to 7.9.

Compuware Shared Services 7.9

CSS Release 7.9 offers the following new features and enhancements:

- Shared Directory Support for Abend-AID Reports

  Abend-AID users can now create/attach multiple DDIO report files to a Shared Directory. The Shared Directory support provides users with a single point of access to all of their report repositories.

- Abend-AID Shared Directory Report Routing Rules Support

  Abend-AID reports can be routed to specific subsets of report databases by specifying routing rules (see the *Abend-AID Installation and Customization Guide* for details) when using the Shared Directory support structure. Based on user-supplied routing criteria (for example, Job Name/Job Accounting Info, etc.), abend reports can be directed to specific subsets of report databases to facilitate customer needs. For example, payroll job abends could be routed to a restricted subset of report databases for security reasons.

- Shared Directory Source Support for All Compiler Languages

  All compiler languages can now be used with the Shared Directory to store/retrieve their Source Listings. This enables Abend-AID source support users to specify a single Source Listing Shared Directory name in place of a source DDIO file to retrieve a source listing, or merge a source listing with a report from a pool of source listing databases. The Shared Directory can be defined with a DBMODEL parameter (or it can be added later), which will allow CSS to dynamically allocate a new database file if all others are full.

  Note: CICS Abend-AID/FX currently does not support Source Shared Directory processing. XPEDITER/TSO releases 7.0, 7.1, and 7.2 allow full Shared Directory support for C language and PL/I LONGNAME support only. XPEDITER/CICS releases 7.4 and 7.5 as well as XPEDITER/TSO 7.0, 7.1, and 7.2 support the use of source databases for all languages as standalone DDIO files only.

- Process Compiler Listings On-The-Fly

  When viewing an Abend-AID report through the Viewing Facility, CSS attempts to merge the source listing with the report. If the source listing cannot be found, the listing can be compiled (preprocessor) into a source DDIO file or an existing compiler listing can be post-processed into a source DDIO file. This enhancement is provided by distributing an exit to two CLISTs (or REXX EXECs) so the customer can write a dialog that will allow a listing to be compiled as described above. Sample dialogs have been supplied to assist in dialog development.

- Abend-AID E-Business Name Changed to Abend-AID for WebSphere MQ

  The Abend-AID E-Business product has been renamed to Abend-AID for WebSphere MQ. All references to E-Business have been replaced in the CSS Viewing Facility.

- Source Support for VisualAge PL/I Abend-AID Reports

  CSS now provides support for customers who view Abend-AID reports via the CSS Viewing Facility to also view VisualAge PL/I program source merged with the Abend-AID report.

- Source Support for Multiple Assembler Programs on the Calling Chain List in the Viewing Facility

  In previous releases, when two or more Assembler programs were linked together, only the abending program would get source support. With this enhancement, all programs on the calling chain can be viewed with their source merged with the Abend-AID report.

- Revise Printed Abend-AID Report So It Matches the Viewing Facility Report

  The printed Abend-AID report has been modified so the format of the report is in the same order as the Viewing Facility.

- Viewing Facility Updated to Display COBOL PIC and VALUE Information from the Working-Storage Section

  In previous releases, the COBOL PIC and VALUE information were available in the stored source listing member. The Viewing Facility has been enhanced to include this information in both the online and printed reports in addition to the stored member.

- Viewing Facility Hardcopy Print Routines Updated to Show the Report to be Processed on the Print Routing Screen

  Previously, when multiple members were selected for printing from the Viewing Facility, the report (or source listing member) was not identified on the print routing screen. This enhancement will display some identifying information each time the routing screen is redisplayed.

- Link to Viewing Facility From Within SDSF When Viewing an Abend-AID Dump

This enhancement enables the user to invoke the Viewing Facility and display the Abend-AID report from within SDSF. The linkage mechanism will place the user directly into the proper report without the need to make note of a job name or report number, and without having to navigate the menus. At the conclusion of the Viewing session, the user will be returned to SDSF.

- Distributed Viewing Support for CICS Abend-AID/FX Transaction Dumps

  Abend-AID customers have the ability to view abend reports and source listings from DDIO files that may be on a remotely connected system. This enhancement provides support for using remote DDIO files to CICS Abend-AID/FX users.

- Cursor-selectable Panel Selection From Within the Viewing Facility

  This enhancement enables the user to tab to certain fields in the Viewing Facility panels, press ENTER, and be routed directly to the appropriate screen for more information on that field.

Enterprise Common Components Release 1.4

Enterprise Common Components 1.4 incorporates the Compuware Shared Services upgrade from Release 7.7 to 7.8.

Compuware Shared Services 7.8

CSS Release 7.8 offers the following new features and enhancements:

- VisualAge PL/I for OS/390 R2.2 Support

  This enhancement provides the capability to use the VisualAge PL/I for OS/390 V2.R2.24 and V2.R2.M1 compiler in the following ways:

  – Preprocess and compile PL/I source code using the new compiler.

  – Postprocess a listing previously generated by the new compiler.

  – Create a DDIO source member closely compatible with source members from earlier PL/I compilers so that it can be accessed and used by other Compuware products that currently use CSS's DDIO source format.

- Support for COBOL Report Writer

  This enhancement supports the COBOL Report Writer Precompiler, PP 5798-DYR with a related run-time library designated PP 5798-DZX. Users may now insert COBOL Report Writer statements into a COBOL program. Use of the input and/or print exit is supported by the Compuware preprocessor.

- Support for COBOL DB/2 Co-processor

  The new COBOL V2R2 compiler features a DB/2 co-processor (requires DB/2 version 7). This replaces the need to use the DB2 preprocessor to translate a COBOL listing that contains "EXEC SQL" statements. The new co-processor is invoked with the 'SQL" compiler option. The CSS COBOL language processor will accept listings compiled from this co-processor and properly create SLS DDIO members with appropriate CLIST entries for the EXEC instructions.

- Support for COBOL 2.2 COMP-5 and 31-digit Decimal Variables

  The new COBOL V2R2 compiler allows packed and zoned decimal data (COMP-5 and DISPLAY) with precision up to 31 digits when the new compiler option ARITH(EXTEND) is specified.

- Show Value of Subscripted Variables in Diagnostics Section (COBOL only)

CSS previously referred viewers to the Working-Storage section whenever there was a subscripted variable (element of an array) in the Diagnostics section. CSS now provides the actual value of the variable whether the subscript is specified as variable, integer, or variable+/-integer. These variables are shown in the “Current values of fields on abending statement” section.

- Remove Sorts from the Assembler Language Processor

  This enhancement removes external sort processing from the CSS Assembler postprocessor and replaces it with a hashing technique. This hashing technique is similar to that used by the PL/I postprocessor, which currently performs no external sorting. Eliminating the sorts provides the following advantages:

  – Performance benefit (CPU and I/O reduction)
  – Reduces size of execution JCL (no sortlib or sort work DD statements)

- Toleration Support for the New Assembler R1.4 Features

  This enhancement allows Assembler options to be specified in an external file. It also provides for use of the two new DC types in High Level Assembler 1.4 for the new 64 bit architecture.

  – FD — double word aligned fixed-point constant
  – AD — double word aligned address constant

- Support for STOP ALL MQ Requests

  This enhancement identifies and flags the new MQSERIES command calls for testing by XPEDITER/CICS users. The existing CLIST table is modified at processing time to handle the new MQ type. CSS returns the statements containing the CALL VERB. Supported call types are: CONNX, BACK, BEGIN, DISC, CLOSE, OPEN, CMIT, CONN, PUT1, SET, XCNVC, _DATA_CONV_EXIT, GET, INQ, and PUT.

- Support for STOP ALL WEB Commands

  This enhancement identifies and flags the new CICS TS 1.3 ‘EXEC CICS WEB typ2” API command calls for debugging WEB-based applications by XPEDITER/CICS users. The existing CLIST table is modified at processing time to handle the new WEB type. CSS returns the statements containing the CALL VERB. The supported call types are: EXTRACT, STARTBROWSE, READNEXT, ENDBROWSE, READ, RECEIVE, WRITE, SEND, and RETRIEVE.

- Reprocess Enhanced Listings

  CSS currently provides the ability to save COBOL compile listings in an “enhanced” format. This enhancement processes these enhanced listings via the CSS postprocessor and stores them in the DDIO file.

- Enhanced Source Support for BL Cells

  Current view-time support properly displays only BLL and BLW cells. As a result, customers receive many ‘DATA ERROR RC=10’ messages for other types of BL cells. In many cases this is caused when Abend-AID does not collect all information at abend time. For the other instances, CSS has improved its support during view-time for the different types of BL cells.

- Upgrade Compuware Base Services/HCI to 4.1

  This enhancement brings the Base Services used by Distributed Viewing to the same release level used by CICS Abend-AID/FX thus ensuring that FX and CSS distribute the same level of Compuware Base Services/HCI.

- Upgrade to HCI 2.3

  This enhancement prevents Host Communication Interface (HCI) version mismatch errors between CSS and CICS Abend-AID/FX. It also changes the communication protocol from LU 6.2 to TCP/IP.

- Shared Directory EXAMINE Command

Depending on the abend activity, it is possible for one or more report files to contain (or not contain) reports listed on the Shared Directory screen. This utility command examines (and optionally fixes) the physical integrity of the Shared Directory, detects out-of-sync conditions, provides a report, and can (optionally) automatically synchronize the report files and Shared Directory.

- Modify SMP/E MCS to Delete Pre-existing CSS Releases

  Previously, when users wanted to install a new CSS release in the same zones as a previous CSS release, miscellaneous link-edit errors occurred because SMP/E attempted to merge the old and new releases (linking them together). This enhancement causes SMP/E to detect that a previous CSS release exists, to automatically delete the old CSS release, then to install the new release. This does not affect users who use separate zones for their CSS releases.

- Save Compiler Listing DSN on Utilities Panel

  CSS saves the name of the dataset containing compiler listings between sessions on the ‘Create JCL - Postprocess Listing’ utilities panel.

- Vocabulary/Appearance Changes

  The following panel changes have been introduced in this release:

  – Improved vocabulary on CSS utility screens to clarify the steps required to use source support.

  – Clearly identifies source and report ‘listings’ on screen prompts.

  – Changed the ‘create JCL for LP preprocessing and postprocessing’ prompt to more accurate terminology.

License Management System Release 2.0

License Management System Release 2.0 includes the following enhancements:

- Enforcement of new Compuware license models on z/900 logical servers executing the z/OS operating system.
- New 14-day grace period allowing Compuware products to run when a logical server exceeds licensed limit or when the name of the logical server doesn’t match the name licensed.
- New 14-day grace period allowing Compuware products to run on a new or upgraded CPU that is not yet licensed.
- New/redesigned License Management System reports.

Introduction

This manual contains instructions for installing, customizing, and maintaining Enterprise Common Components (ECC).

Enterprise Common Components is essentially a packaging method for the following Compuware facilities:

- License Management System (LMS)
- Compuware Shared Services (CSS)
- Base Services
- Host Communications Interface (HCI)

Manual Structure

This *Enterprise Common Components Installation and Customization Guide* is composed of the following chapters:

Chapter 1, "ECC Overview": A description of the individual components that make up Enterprise Common Components with overview information on each of the parts. These include: ECC, CSS, LMS, Base Services, and HCI.

Chapter 2, "Preparing for ECC Installation": Critical information for planning your installation of ECC including the LMS, CSS, Base Services, and HCI components. This chapter also lists which components are required based on your Compuware product, and also lists the libraries that are created during installation with SMP/E.

Chapter 3, "Installing Enterprise Common Components": Step-by-step instructions for installing ECC.

Chapter 4, "Applying Maintenance": Mandatory step-by-step instructions for applying maintenance to your installation of ECC.

Chapter 5, "CSS Customization": Step-by-step instructions for customizing your installation of CSS and the Compuware Viewing Facility (Compuware/VF).

Chapter 6, "LMS Customization": Step-by-step instructions for setting up and customizing LMS for use with the Compuware products licensed by your organization.

Chapter 7, "HCI Customization": Step-by-step instructions for setting up and customizing HCI for use with the Compuware products licensed by your organization.

Chapter 8, "HCI Facilities": A description of the facilities available to configure HCI for individual product use.

Intended Audience

This manual is intended for installers and application programmers. It is only to be used for installing the MVS/ESA, OS/390, and z/OS versions of Compuware products.

Related Publications

For more information about CSS, LMS, Base Services, or HCI, refer to the Compuware documentation shipped with the ECC tape. For more information about Distributed Viewing Support, refer to the *Abend-AID Installation and Customization Guide* or *Abend-AID for CICS Installation and Customization Guide*. Instructions for accessing this documentation are given in the following paragraphs. For more information on compiling COBOL, PL/I, Assembler, or C/C++ programs, refer to the appropriate IBM manuals.

FrontLine Support Website

Access online technical support for Compuware products via our FrontLine support website. View or download documentation, frequently asked questions, and product fixes, or directly e-mail Compuware with questions or comments. To access FrontLine, you must first register and obtain a password at **http://frontline.compuware.com**.

Online Documentation

Documentation for this product is provided on CD-ROM in several electronic formats.

- View PDF files with the free Adobe Acrobat Reader, available at **http://www.adobe.com**.
- View HTML files with any standard Web browser.
- View BookManager softcopy files with any version of IBM BookManager READ or the IBM Softcopy Reader. To learn more about BookManager or to download the free Softcopy Reader, go to **http://www.ibm.com**.

World Wide Web

Compuware's site on the World Wide Web provides information about Compuware and its products. The address is **http://www.compuware.com**.

Notation Conventions

This section describes common terms and syntax conventions used in this manual. A change bar (|) indicates an update to the manual for this release.

Terminology

The following terms are defined as they are used in this manual:

- **DDIO (Dump Dataset Input Output) file** is used generically to mean any of the following types of files:
  - Report File
  - Source Listing File
  - Transaction Databases & Shared Directories
  - Report Shared Directory & Database
  - Source Listing Shared Directory
  - Source Listing Database

  For a list of the types of DDIO files created by the CSS Utilities and which products use/require them, refer to the *Compuware Shared Services User/Reference Guide*.
- **Member** is used to identify an individual diagnostic report in an Abend-AID report file, a source listing in a source listing file, or a profile in a profile file.

- **Entry** is used to identify an individual transaction diagnostic report in a transaction database for Abend-AID for CICS.

Getting Help

Compuware continually strives to improve our software products and documentation. Feedback from our customers helps us to maintain our quality standards.

While the software should execute according to documented specifications, there are times when problems do occur. If problems arise, check the Compuware manuals for assistance. You may also need to consult your site's technical representative for Compuware assistance.

If you have attempted to solve the problem using the resources listed above and the difficulty persists, contact the Compuware Technical Support Hotline for assistance. Technical support is also available via our FrontLine Support web site as described in "Related Publications".

Please have the following information available before calling. This information will assist the Technical Support personnel in determining the exact cause of your problem in a timely manner.

Note: If you encounter a CSS problem when compiling your program, Compuware Technical Support staff may need additional information. You can obtain this information by using the CSS Problem Documentation Utility. Refer to the "Using the Problem Documentation Utility" appendix in the *Compuware Shared Services User/Reference Guide* for complete instructions.

1. Use the following sample JCL members to identify the PTFs and/or APARs that have been applied to the respective ECC components' libraries:
   - CXLPTFS or CXLZAPS in the CSS SLCXCNTL library
   - $LMLPTFS in the LMS SLMSCNTL library
   - $DVLPTFS in the Base Services SKMPCNTL library
   - $HCLPTFS in the HCI SLHCCNTL library

   A lengthy SMP/E report will be generated with exceptions to the CXLZAPS job. That will produce a summarized version of the PTF listing.
2. Determine the release number of the ECC-enabled product.
3. For CSS problems, identify the compiler language and the release number being processed.
4. Identify the CICS release in use if the problem involves a CICS product.
5. Identify the operating system release to help determine operating system dependencies.
6. If an abend occurs, note the displacement and the module in which the interrupt occurs.
7. Note the sequence of issued commands that resulted in the abend, the data type, and the programming language being used.
8. Locate your Compuware customer number.

Compuware Technical Support
Compuware Corporation
One Campus Martius
Detroit, MI 48226-5099
1-800-538-7822

http://frontline.compuware.com

Chapter 1.
ECC Overview

This chapter explains the composition of Compuware's Enterprise Common Components (ECC) and provides overview information on each of its component parts.

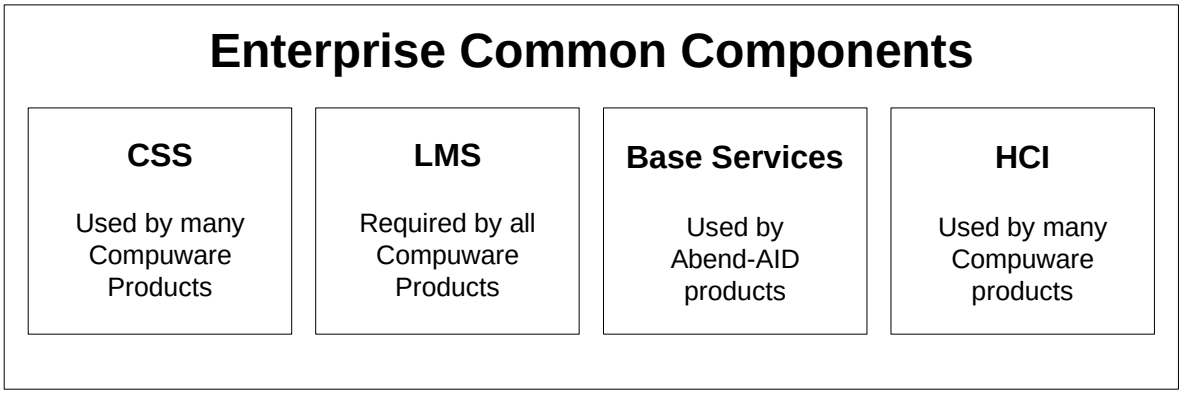
What is ECC?

Enterprise Common Components is the packaging method for the following Compuware facilities:

- Compuware Shared Services (CSS)
- License Management System (LMS)
- Base Services
- Host Communications Interface (HCI)

Which parts of ECC you need to install depends on which Compuware products are licensed by your organization and which parts of ECC have already been installed. All Compuware products require the License Management System. Many Compuware products utilize Compuware Shared Services to provide crucial core functionality. Base Services and HCI are used by Abend-AID Release 9.0 and more current for their Distributed Viewing Support feature. Abend-AID for CICS also requires the Base Services and HCI components. A table is provided in **Chapter 2, "Preparing for ECC Installation"** that makes it easy to determine exactly which parts of ECC you need to install.

Figure 1-1. ECC Composition



Each of the following components are installed using SMP/E and can be installed individually, depending on your site's requirements:

- **Compuware Shared Services (CSS):** An enterprise-wide tool for sharing compiled program information, CSS is utilized by many Compuware products spanning the Fault Management and Enterprise Testing Solutions product lines. For more information, see "CSS Overview" on page 1-2.

- **License Management System (LMS):** Using LMS enables you to manage access to all the Compuware products used by your organization. LMS consists of several small components which allow you to establish, maintain, diagnose, and upgrade your access to the Compuware products licensed by your enterprise:
  - An ISPF License Administration Utility application
  - A runtime environment
  - A program interface to the runtime environment employed by Compuware products.

  LMS License Certificates are English-like text files that are sent to you electronically. You will receive a License Certificate for each new product release under LMS. You can centrally manage all of your organization's License Certificates in a single License File that can then be distributed to various sites.
- **Base Services:** This component allows Abend-AID or Abend-AID for CICS users to merge Abend-AID reports and source listings that are on different MVS images with unshared DASD or to access the files individually. Users access the files through the Compuware Viewing Facility or the Abend-AID for CICS viewing server. Working from individual local systems, any number of users can access reports and source listing files that reside locally and/or remotely on other MVS images. Distributed Viewing Support is available beginning with Abend-AID 9.0 and more current.
- **Host Communications Interface (HCI):** The Host Communications Interface is a facility that provides connectivity between mainframe-based programmer productivity software and peer-node software running on other platforms in a network. The HCI also serves as a server activation facility that is responsible for starting and monitoring application programs that are using the communications facilities.

  Note: Base Services and HCI are required components for Abend-AID for CICS users. Abend-AID users only require these components if they are using Distributed Viewing Support.

CSS Overview

Compuware Shared Services (CSS) is an integral component of many Compuware products spanning the Fault Diagnosis, Interactive Analysis and Debugging, and File and Data Management product lines. CSS provides storage, retrieval, and maintenance for Abend-AID reports, source listings, and transaction reports on datasets called DDIO files. Shared directories and their attached databases are also types of DDIO files being utilized for more efficiency in maintaining information and activity within these files. CSS also provides language processing support for COBOL, PL/I, VisualAge PL/I, Enterprise PL/I, Assembler, C for OS/390, and C for MVS/ESA V3 R2.

The following products currently use CSS.

- Abend-AID
- Abend-AID for CICS
- Abend-AID Fault Manager
- Abend-AID for WebSphere MQ
- XPEDITER/CICS
- XPEDITER/TSO and XPEDITER/IMS
- XPEDITER/Code Coverage
- STROBE

Benefits of CSS

Prior to CSS Release 1.0, the components that make up CSS were distributed within each product library. For customers with multiple Compuware products, the result was a high degree of redundancy within their product libraries and a complex system of product maintenance. By combining components used by several products into a single unit, CSS provides:

- **Easier Installation**

  All CSS components can be installed at one time rather than piecemeal during installation of other Compuware products.

- **Simplified Maintenance**

  All maintenance to CSS components is applied to ONE set of libraries. It is no longer necessary to search through each product library, determine if a fix must be applied, and then apply the same fix to multiple libraries.

- **Frequent New Release Updates**

  New release updates are made available more frequently.

- **Disk Storage**

  CSS programs and subroutines reside in a separate library. Compuware customers with multiple products installed can recover disk space previously used by duplicate modules and programs using CSS.

- **Fewer JCL Changes**

  The need to change CICS startup JCL, TSO logon PROCs, compile PROCs, and/or rundecks is reduced or eliminated because a central library exists for all products using CSS. This helps to simplify installation.

- **Separate Documentation**

  Documentation for common components is no longer imbedded in the individual product manuals. In addition to your product documentation, you also receive the *Enterprise Common Components Installation and Customization Guide* in printed form, and the *Compuware Shared Services User/Reference Guide* and *Enterprise Common Components Messages and Codes* on CD. You no longer need to hunt through the product manuals for information on formatting DDIO files, printing reports and listings, and setting the parameters to run the language processors. Additionally, this ECC documentation set provides greater detail on processing options and customization than was previously available.

System Environment

CSS supports CICS Releases 1.7 through 4.1. It also supports CICS Transaction Server for OS/390 Release 1.3 and earlier and all releases of IBM MVS, as well as ISPF/PDF Versions 2.0 or more current.

The following is a list of compilers supported by the language processors:

- Assembler
  - High Level (HLASM) Assembler 1.1, 1.2, 1.3, 1.4
  - Assembler H
  - Assembler F
- COBOL
  - Enterprise COBOL for z/OS and OS/390 Release 3.1.0
  - COBOL for OS/390 and VM 2.1, 2.1.1, and 2.2 (MLE Support)
  - COBOL for MVS and VM 1.2 and 1.2.2 (MLE Support)
  - AD/Cycle COBOL/370 1.0, 1.1

  - VS/COBOL II 1.2, 1.3, 1.4
  - OS/VS COBOL 2.4
  - COBOL for z/OS 3.2

    Note: Compuware supports the MVS portion of the IBM compilers COBOL for MVS and VM, and COBOL for OS/390 and VM.

  - CA-OPTIMIZERs
    - CA-OPTIMIZER II 1.1, 1.2, 1.3, 2.0, 2.1, 2.2, 3.0
    - CA-OPTIMIZER III 5.0, 5.1, 6.0, 6.1, 7.0

    Note: CA-OPTIMIZER II Release 2.0 supports COBOL/370.

- PL/I
  - Enterprise PL/I R3.1
  - VisualAge PL/I R2.2
  - PL/I for MVS and VM 1.1.1
  - AD/Cycle PL/I 1.1
  - PL/I 1.5.1, 2.2, 2.3, 3.2

  Note: Compuware supports the MVS portion of the IBM compilers PL/I for MVS and VM 1.1.1.

- C
  - C for MVS/ESA V3 R2
  - OS/390 C releases through V2 R10

CSS Components

CSS consists of the following components:

- Compuware common files and utilities
  - DDIO files, Shared Directories and associated Report, Transaction, and Source Listing Databases
  - Batch file utilities: CWDDSUTL, CWFXSDUT, CWAASDUT, and CWDDLPUT
- Compuware Viewing Facility (Compuware/VF)
- Compuware Language Processors (COBOL, Assembler, PL/I, C, VisualAge PL/I, and Enterprise PL/I)
- Security Exit Program

Common Files and Utilities

CSS components include common files and utilities that provide storage, retrieval, and maintenance functions.

DDIO Files

A proprietary file access method, called DDIO, that stores Abend-AID reports, transaction reports, and source listings for several Compuware products. DDIO files can be allocated as either VSAM or sequential datasets. Different products may require different types of DDIO files.

For detailed information on allocating and formatting DDIO files, see the “Allocating and Formatting DDIO Files” chapter in the *Compuware Shared Services User/Reference Guide.*

CSS Utilities and Batch File Utilities

CSS Utilities walks you step-by-step through CSS language processor functions and DDIO file manipulation using easy-to-use panels.

The batch file utilities (CWDDSUTL, CWAASDUT, and CWDDLPUT) provide the same functions as CSS Utilities in a batch environment.

For detailed information on these utilities, refer to the "CSS Utilities", "Batch File Utility CWDDSUTL", "Batch File Utility CWFXSDUT", Batch File Utility CWAASDUT, and "Batch File Utility CWDDLPUT" chapters in the *Compuware Shared Services User/Reference Guide.*

Compuware Viewing Facility (Compuware/VF)

The Compuware Viewing Facility (Compuware/VF) provides immediate, online, menu-driven access to reports and source listings. From Compuware/VF, you can view a directory that summarizes the members in a DDIO file. You can then access the source listing or Abend-AID report from the directory. While viewing an Abend-AID report, you can directly access any available section of the report, search for particular date or currency formats within the report, or link to Compuware File-AID (to use the BROWSE or EDIT functions only).

Note: To access File-AID from Compuware/VF, one of the following File-AID releases must be installed: File-AID/MVS 8.5 or more current, File-AID for DB2 V3R9 or more current, or File-AID for IMS 4.7 with product update F1. In addition to one of the File-AID products, Abend-AID 9.1 or more current must also be installed.

Compuware/VF has many features that enable you to view DDIO files easily and quickly. Its navigation commands let you directly access desired report sections to eliminate cumbersome scrolling. From Compuware/VF you can view, print, lock, unlock, and delete members in a DDIO file online. The members can easily be manually locked to ensure that they remain stored on the DDIO file until you unlock them. Compuware/VF also lets you print members at view time.

Compuware/VF enables you to access the following types of DDIO files:

- Abend-AID reports
- Abend-AID, Abend-AID for CICS, XPEDITER/TSO, and XPEDITER/CICS source listings.

  Note: Viewing access to Abend-AID for CICS source listing files is also provided by the Abend-AID for CICS view server. Abend-AID for CICS users do not need to install Compuware/VF if a server has already been configured to support viewing access. Compuware/VF may be installed, however, in order to view other products' source listing files.

For more information on the Viewing Facility, refer to the "Compuware Viewing Facility" chapter in the *Compuware Shared Services User/Reference Guide.*

CSS Language Processors (LP)

The language processors capture information about a compiler listing and store it in a DDIO file. The majority of this information is gathered from the compiler listing, but optionally, SYSIN and SYSLIB are also examined.

A language processor can be run using two methods:

- preprocessor
- postprocessor

You may use the preprocessor or postprocessor (whichever is best for your site as needed.) For information about the pre- and postprocessors, refer to the appropriate language processor chapter in the Compuware Shared Services User/Reference Guide.

Language Processor Types

CSS provides language processing support for the following languages:

- COBOL
- PL/I, VisualAge PL/I, Enterprise PL/I
- Assembler
- C

The COBOL language processor reads the compiler listing produced by the COBOL compiler and writes either a compiler listing or an enhanced listing to the DDIO file. The enhanced listing merges the information normally found in the data division map (DMAP) and the condensed listing (CLIST) or procedure map (PMAP) into the source statement lines. The COBOL compiler options in effect are sorted alphabetically and appear at the beginning of the enhanced listing. The enhanced listing provides you with a condensed hardcopy of the compiler information.

The COBOL and PL/I listings may be used to provide source support during a debugging or diagnostic session using any of the following products:

- Abend-AID
- Abend-AID for CICS
- XPEDITER/CICS
- XPEDITER/TSO

The Assembler listings may be used to provide source support during a debugging or diagnostic session using any of the following products:

- Abend-AID
- XPEDITER/CICS
- XPEDITER/TSO

The C language processor reads the compiler listing produced by the C compiler and writes the compiler listing to the DDIO file and SYSCPRT. The source listing in DDIO can then be used to provide source support during a debugging or diagnostic session using:

- XPEDITER/TSO

Language support for Enterprise PL/I and VisualAge PL/I is provided for the following products:

- Abend-AID
- XPEDITER/CICS
- XPEDITER/TSO

Security Exit Program

CSS also includes the Compuware Security Exit Program. This is an optional user-written exit for your Compuware products. It can be used in conjunction with your existing security package to secure sensitive data used by the following Compuware products:

- Abend-AID (VisualAge PL/I and Enterprise PL/I language support is also available)
- Abend-AID Fault Manager
- Abend-AID for WebSphere MQ
- Abend-AID for CICS
- STROBE
- XPEDITER/CICS (VisualAge PL/I and Enterprise PL/I language support is also available)
- XPEDITER/Code Coverage

- XPEDITER/TSO and XPEDITER/IMS (VisualAge PL/I and Enterprise PL/I language support is also available)

The Compuware Security Exit enables you to control access to DDIO file members or restrict access to various commands through a user-written exit program. For the Abend-AID for CICS product, you can restrict access to some of the CWFXSDUT functions for transaction database and source listing files. For transaction databases, you can use the Security Exit Program to restrict access to individual dumps within a transaction report database.

If you install a Security Exit program, it will be called by all Compuware products installed at your site. For more information on the Security Exit Program, see the "CSS Security Exit" chapter in the *Compuware Shared Services User/Reference Guide*.

Executing CSS

CSS is executed at the following times:

- Abend-AID and Abend-AID for CICS
  - at compile time (when source support is used)
  - at time of abend or application program failure
  - at view time when using Compuware/VF or the Abend-AID for CICS Viewer
  - at installation and maintenance time when formatting a DDIO file.
  - when a compile listing that has been stored elsewhere is needed for Abend-AID or Abend-AID for CICS viewing with source support.
- XPEDITER/TSO and XPEDITER/CICS
  - at compile time (when source support is used)
  - at execution time for interactive debugging
  - at view time when using Compuware/VF
  - at installation and maintenance time when formatting a DDIO file.

LMS Overview

LMS is located on the ECC tape included in the product shipping package. LMS enables you to centrally administer Compuware's product License Certificates and manage access to Compuware products at your site. LMS includes several components that together enable you to establish, maintain, diagnose, and upgrade access to the Compuware products licensed by your enterprise. LMS replaces the customer profile utility provided with previous versions of Compuware mainframe products.

LMS Advantages

LMS provides the following advantages:

- Electronically-delivered License Certificates that contain product access parameters in a human-readable text format
- Programs and reports to verify the contents of the License File and LMS runtime environment
- Capability to employ a single centralized Enterprise License File and centralized administration of License Files
- Easy updates to License Certificates and the License File with no disruption to your Compuware products
- No production impact from the testing of LMS runtime systems, installation and validation of new License Certificates, or installation of new or updated LMS software

- Optional E-mail alerts and SMF logging when licensing events occur.

Operating Environment

LMS operates under IBM MVS/ESA Release 4.3 and more current, and all OS/390 and z/OS releases.

The License Management Process

The license management process begins when you acquire a Compuware product through a license agreement, trial agreement, or beta agreement.

Information Gathering

Compuware obtains the relevant information for license management from you at the time of the agreement. Your organization supplies information such as name and the sites and CPUs for which the product is being licensed. Information about the Compuware product licensed, such as product name, release number, and options licensed, is obtained from your Compuware representative.

License Certificate

Upon the completion of an agreement, Compuware creates a License Certificate representing a portion of the information in that agreement. The License Certificate is used by LMS to provide access to Compuware products. The License Certificate is **not** the same thing as a license agreement. You are still responsible for abiding by your license agreement, and although not its primary role, you will find that the LMS can help in that effort.

The License Certificate, LMS, and the Compuware product are all delivered to you by Compuware. Typically, you receive the License Certificate via e-mail, but other methods can be used, if necessary.

Installation

The first time your enterprise receives LMS, you must install the LMS software. Then you set up your LMS environment, import your License Certificate into a **License File**, activate the **Runtime License Management System**, and install your Compuware product. From then on, access to your Compuware mainframe products occurs transparently.

After you accomplish the initial LMS installation, you would typically only need to import a new product License Certificate and re-initialize the LMS runtime environment in conjunction with events such as the following:

- obtaining a new release of a product under the terms of your software maintenance agreement
- adding new products through additional license agreements
- amending your agreement to include new options
- changing the CPUs licensed in your original agreement.

Base Services Overview

Base Services and HCI are required to use Abend-AID Distributed Viewing Support (DVS). Abend-AID or Abend-AID for CICS users can use DVS to view reports whether residing on a local or remote system. Base Services is also used by Abend-AID for CICS during regular viewing.

Abend-AID users can view both merged and base reports where either the base report, source listings, or both reside on a remote system. Access to the remote files is provided through the optional Distributed Viewing Support component.

With Distributed Viewing Support, Abend-AID users can utilize source support across remote MVS systems that do not share DASD. Working from individual local systems, any number of users can access report and source listing files that reside locally and/or remotely on other MVS images.

Distributed Viewing Support enables you to access the following types of DDIO files:

- Abend-AID reports
- Abend-AID for CICS dump reports
- Abend-AID, Abend-AID for CICS, XPEDITER/TSO, and XPEDITER/CICS source listings.

Base Services and HCI are distributed with CSS on the Enterprise Common Components distribution tape. Instructions for configuring Distributed Viewing Support and setting up the servers are provided in the Abend-AID and Abend-AID for CICS product documentation. Instructions for installing Base Services and HCI can be found in the *Enterprise Common Components Installation and Customization Guide*.

Distributed Viewing Support is available beginning with Abend-AID Release 9.0 and CICS Abend-AID/FX (now Abend-AID for CICS) release 4.4.

Note: Viewing access to Abend-AID for CICS source listing files is also provided by the Abend-AID for CICS **view** server. Abend-AID for CICS users do not need to install Compuware/VF if a server has already been configured to support viewing access. Compuware/VF may be installed, however, in order to view other products' source listing files.

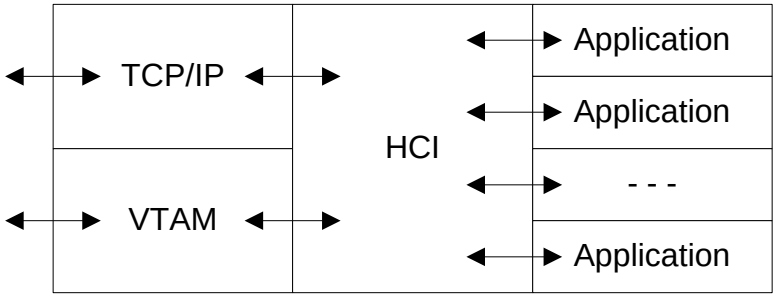
HCI Overview

The Host Communications Interface (HCI) is a facility that provides connectivity between mainframe-based programmer productivity software and peer-node software running on other platforms in a network. These communications facilities provide a simple application programming interface so that Compuware software can be written to communicate with peer programs running on different platforms. Additionally, the communications system is flexible, simple to operate, and reliable. It has been designed to support open-ended communications protocols, and it can operate on older releases of the MVS operating system. It allows application programs to communicate over any one of several protocols without knowing which protocol is in use at any given time.

The HCI also serves as a server activation facility that is responsible for starting and monitoring application programs that are using the communications facilities.

And finally, the HCI has been designed with no architected upper limit on the number of concurrent sessions supported, nor on throughput rates. The following figure shows how the HCI fits into the scheme of the TP application and the MVS subsystem components.

Figure 1-2. TP Subsystem, HCI and Application. HCI Position as Middleware



For more information about Host Communications Interface features, refer to **Chapter 8, "HCI Facilities"**.

Chapter 2.
Preparing for ECC Installation

This chapter contains important information that can help you prepare to install the Enterprise Common Components (ECC) tape in the most efficient manner. Review it before beginning the installation process.

Required Components and Procedures

This section explains how to determine which parts of ECC need to be installed at your site and which chapters in this manual contain the applicable installation and customization procedures.

Component Selection

Different Compuware products utilize different portions of ECC. Use Table 2-1 to determine which parts of ECC are needed at your site.

Notice that Compuware's License Management System (LMS) is required for every product, while Base Services and HCI (which are required for Abend-AID Distributed Viewing Support) are optional components.

DVS can be used with the following products:

- Abend-AID version 9.0 or later
- CICS Abend-AID/FX version 4.4 or Abend-AID for CICS version 4.5 or later

Table 2-1. Components Utilized by Each Compuware Product

Compuware Product	LMS	CSS	Base Services	HCI
Abend-AID	x	x	x *	x*
Abend-AID Fault Manager	x	x		
Abend-AID for CICS	x	x	x	x
DBA-XPERT for DB2	x			
File-AID/Data Solutions	x			
File-AID for DB2	x			
File-AID for IMS	x			
File-AID/MVS	x			
File-AID/RDX	x			
*QA*Batch	x			
*QA*Hiperstation	x			
*QA*Playback	x			
STROBE	x	x**		x
UNIFACE	x			x
XPEDITER/CICS	x	x		

Table 2-1. Components Utilized by Each Compuware Product

Compuware Product	LMS	CSS	Base Services	HCI
XPEDITER/Code Coverage	x	x		
XPEDITER/DevEnterprise	x	x		x
XPEDITER/TSO and IMS	x	x		
XPEDITER/Xchange	x			
* Base Services and HCI are required only if utilizing Abend-AID Distributed Viewing Support.				
** CSS is required only if creating profiles with source code indexing. CSS minimum release is 7.8.				

Required Procedures

Determining which chapters to use in this manual is actually very simple:

- Every mainframe Compuware product utilizes LMS. Therefore, every installation of a Compuware product requires, at a minimum, that the latest maintenance be applied to LMS. So regardless of which product you are installing, you must perform the applicable procedures in the following chapters:
  - **Chapter 3, "Installing Enterprise Common Components"** (if current release has not been previously installed)
  - **Chapter 4, "Applying Maintenance"**
  - **Chapter 6, "LMS Customization"**
- If the Compuware product you are installing requires CSS, you must also perform the procedures in **Chapter 5, "CSS Customization"**. To configure Distributed Viewing Support, see the *Abend-AID Installation and Customization Guide* or the *Abend-AID for CICS Installation and Customization Guide.*
- If you are installing STROBE, UNIFACE, or XPEDITER/DevEnterprise, you must also follow the procedures in **Chapter 7, "HCI Customization"** and optionally **Chapter 8, "HCI Facilities"**.

The information within each chapter will direct you as to which steps apply in your situation.

ECC Packaging

The ECC product tape, which includes LMS, CSS, Base Services, and HCI, is packaged for installation using System Modification Program Extended (SMP/E). SMP/E is a method developed by IBM to install software products in the MVS environment.

SMP/E provides enhanced installation management and maintenance tracking capabilities. To facilitate SMP/E, the LMS, CSS, Base Services, and HCI components are shipped on the ECC tape separately from the products. The separate ECC tape contains both the base code and all cumulative, approved maintenance to-date. The ECC tape is shipped with each Compuware product that utilizes the CSS, Base Services, HCI, or License Management components. To install the ECC tape, refer to **Chapter 3, "Installing Enterprise Common Components"**.

A separate tape containing **only** the cumulative ECC maintenance is available on request. You can obtain the maintenance tape by calling the Compuware Technical Support Hotline (1-800-538-7822). This maintenance tape contains all maintenance approved to-date. To install the ECC maintenance tape to an existing installation, you only need to perform the steps described in **Chapter 4, "Applying Maintenance"**. The separate maintenance tape allows you to receive upgrades quickly — without installing a new

release. It is recommended that you periodically request and apply the maintenance tape to keep your product version current.

The installation utilizes the Installation JCL Customization Facility consisting of a set of REXX EXECs, JCL skeletons, and panels. This facility prompts you for the installation information and then builds the necessary jobs required to perform the SMP/E installation. You can also view reporting information from the installation facility, such as parameters and execution information.

SMP/E Installation and Maintenance

This section describes allocating SMP/E zones, SMP/E jobs to be executed, and libraries created during the SMP/E installation.

Allocating SMP/E Zones

SMP/E uses a VSAM cluster called a CSI (Consolidated Software Inventory) to control all product installation and maintenance information. Within a CSI, there are a minimum of 3 zones:

- **Global** zone
- **Target** zone
- **Distribution** zone

If a CSI is shared among several different products, there will be a single Global zone and a separate set of Target and Distribution zones for each product installed as illustrated in Table 2-2. A site may have many CSIs, each with its own Global, Target, and Distribution zones.

CAUTION:

To prevent possible conflicts, Compuware strongly recommends that all Compuware products and components distributed via SMP/E share a single Global zone reserved solely for Compuware products. Target and distribution zones should not be shared with other products or different releases of the same product.

If you need to maintain more than one release of CSS at a time under SMP/E control, then they must be installed into separate Target and Distribution Zones. If you try to install CSS into the same Target and Distribution Zones as an existing release, then the previous release of CSS will be deleted.

Table 2-2. SMP/E Zones

Global Zone	
Target Zone Product 1	Distribution Zone Product 1
Target Zone Product 2	Distribution Zone Product 2
Target Zone Product 3	Distribution Zone Product 3

There are many different ways to allocate the zones. These methods include the following:

- The Global, Target, and Distribution zones may all reside together in a **single** VSAM cluster.
- The zones may each reside in **separate** VSAM clusters.
- The zones may each reside in a **combination** of VSAM clusters.

For example, you could have a single VSAM cluster for your Global zone and combine all the Target and Distribution zones in a second VSAM cluster. Or each set of Target and Distribution zones may be combined in a single VSAM cluster. Compuware recommends that all zones be placed in separate VSAM clusters so that each zone can be backed up and restored individually, if necessary, without impacting any other zone.

Installing with SMP/E

The SMP/E portion of the installation process for LMS, CSS, Base Services, and/or HCI consists of four basic steps:

1. Allocate and initialize the SMP/E datasets
2. Receive base and maintenance
3. Apply base and maintenance
4. Accept base and maintenance

After a function or maintenance (such as a PTF) has gone through the ACCEPT step, that becomes your current release-base level. At any time prior to the ACCEPT step, you may issue a RESTORE command to reset your Target libraries to the way they were as of the last ACCEPT command. The RESTORE command is used to remove maintenance, or even entire functions, while the ACCEPT command effectively sets a new base level of the release to which further maintenance will be applied. For more information about using the RESTORE command, refer to your site's IBM SMP/E documentation.

Installing ECC Using JCL Customization Facility

The Installation JCL Customization Facility consists of a set of REXX EXECs, JCL skeletons, messages, and panels contained in the SMP/E installation sample library. The facility will display several panels to obtain installation information, and then will build the necessary jobs required to perform the SMP/E installation. The facility allows you to input SMP/E information, such as the Global SMP/E and SMP/E dataset high-level qualifiers, one time. They can then be used in all SMP/E installation jobs.

Note: Compuware recommends a minimum of ISPF/PDF version 3.5 when using the Installation JCL Customization Facility. REXX is also required.

Using the Installation JCL Customization Facility, you can install any or all of the following:

- Compuware Shared Services (CSS)
- License Management System (LMS)
- Base Services
- Host Communications Interface (HCI)

Please note the following:

- You must install the CSS component; however, the Base Services and HCI components must be installed only if you are using the following products:
  - Abend-AID 9.0 or later with Distributed Viewing Support
  - CICS Abend-AID/FX 4.4 or Abend-AID for CICS 4.5 or later
- If you require only CSS, install only the CSS component.
- If you require only HCI, install only the HCI component.
- If you need to install more than one component, Compuware recommends simultaneously installing all components.

ECC Component Prefixes and FMIDs

Compuware has registered the following element prefixes with IBM for the ECC components:

- **LCX** for CSS
- **LMS** for LMS
- **KMP** for Base Services
- **LHC** for HCI
- **KCW** a generic Compuware prefix.

The FMIDs for each component are listed below.

- MLCXnnn - CSS base code
- MLMSnnn - LMS base code
- NLCXnnn - CSS Japanese support
- MKMPnnn - Base Services base code
- OKMPnnn - Base Services Japanese support.
- MLHCnnn - HCI base code

Libraries Created During SMP/E Installation

The chart in Table 2-3 lists the libraries created during the ECC installation using SMP/E.

Table 2-3. Libraries Created During Installation With SMP/E

DDname	Library Type and Content	Dataset Name as Distributed
INSTLIB	ECC SMP/E Sample Library	COMPWARE.KCWnnn.INSTALL
SMPCSI	Compuware Global CSI	COMPWARE.GLOBAL.CSI
SMPLOG	Compuware SMP/E System File	COMPWARE.GLOBAL.SMPLOG
SMPPTS	Compuware SMP/E System File	COMPWARE.GLOBAL.SMPPTS
ALMSSAMP	LMS Distribution Sample Library	COMPWARE.MLMSnnn.ALMSSAMP
ALMSLOAD	LMS Distribution Load Library	COMPWARE.MLMSnnn.ALMSLOAD
ALMSCNTL	LMS Distribution Sample/CLIST Library	COMPWARE.MLMSnnn.ALMSCNTL
ALMSMENU	LMS Distribution Messages	COMPWARE.MLMSnnn.ALMSMENU
ALMSPENU	LMS Distribution Panels	COMPWARE.MLMSnnn.ALMSPENU
ALMSSENU	LMS Distribution ISPF Skeletons	COMPWARE.MLMSnnn.ALMSSENU
LMSnnnD	LMS Distribution Zone	COMPWARE.MLMSnnn.DZONE.CSI
LMSnnnT	LMS Target Zone	COMPWARE.MLMSnnn.TZONE.CSI
SLMSSAMP	LMS Target Sample Library	COMPWARE.MLMSnnn.SLMSSAMP
SLMSLOAD	LMS Target Load Library	COMPWARE.MLMSnnn.SLMSLOAD
SLMSCNTL	LMS Target Sample/CLIST Library	COMPWARE.MLMSnnn.SLMSCNTL
SLMSMENU	LMS Target Messages	COMPWARE.MLMSnnn.SLMSMENU
SLMSPENU	LMS Target Panels	COMPWARE.MLMSnnn.SLMSPENU
SLMSSENU	LMS Target ISPF Skeletons	COMPWARE.MLMSnnn.SLMSSENU
SMPMTS	LMS SMP/E System File	COMPWARE.MLMSnnn.SMPMTS
SMPSCDS	LMS SMP/E System File	COMPWARE.MLMSnnn.SMPSCDS
SMPSTS	LMS SMP/E System File	COMPWARE.MLMSnnn.SMPSTS
ALCXCNTL	CSS Distribution Sample Library	COMPWARE.MLCXnnn.ALCXCNTL
ALCXEXEC	Distribution REXX Library	COMPWARE.MLCXnnn.ALCXEXEC

Table 2-3. Libraries Created During Installation With SMP/E

DDname	Library Type and Content	Dataset Name as Distributed
ALCXLOAD	CSS Distribution Load Library	COMPWARE.MLCXnnn.ALCXLOAD
ALCXMENU	CSS Distribution English Messages (ISPMLIB)	COMPWARE.MLCXnnn.ALCXMENU
ALCXMJPN[1]	CSS Distribution Japanese Messages (ISPMLIB)	COMPWARE.MLCXnnn.ALCXMJPN
ALCXPENU	CSS Distribution English Panels (ISPPLIB)	COMPWARE.MLCXnnn.ALCXPENU
ALCXPJPN[1]	CSS Distribution Japanese Panels (ISPPLIB)	COMPWARE.MLCXnnn.ALCXPJPN
LCXnnnD	CSS Distribution Zone	COMPWARE.MLCXnnn.DZONE.CSI
LCXnnnT	CSS Target Zone	COMPWARE.MLCXnnn.TZONE.CSI
SLCXCNTL	CSS Target Sample Library	COMPWARE.MLCXnnn.SLCXCNTL
SLCXEXEC	CSS Target REXX Library	COMPWARE.MLCXnnn.SLCXEXEC
SLCXLOAD	CSS Target Load Library	COMPWARE.MLCXnnn.SLCXLOAD
SLCXMENU	CSS Target English Messages (ISPMLIB)	COMPWARE.MLCXnnn.SLCXMENU
SLCXMJPN[1]	CSS Target Japanese Messages (ISPMLIB)	COMPWARE.MLCXnnn.SLCXMJPN
SLCXPENU	CSS Target English Panels (ISPPLIB)	COMPWARE.MLCXnnn.SLCXPENU
SLCXPJPN[1]	CSS Target Japanese Panels (ISPPLIB)	COMPWARE.MLCXnnn.SLCXPJPN
SMPMTS	CSS SMP/E System File	COMPWARE.MCXnnn.SMPMTS
SMPSCDS	CSS SMP/E System File	COMPWARE.MLCXnnn.SMPSCDS
SMPSTS	CSS SMP/E System File	COMPWARE.MLCXnnn.SMPSTS
AKMPCNTL	Base Services Distribution Sample Library	COMPWARE.MKMPnnn.AKMPCNTL
AKMPLOAD	Base Services Distribution Load Library	COMPWARE.MKMPnnn.AKMPLOAD
KMPnnnD	Base Services Distribution Zone	COMPWARE.MKMPnnn.DZONE.CSI
KMPnnnT	Base Services Target Zone	COMPWARE.MKMPnnn.TZONE.CSI
SKMPAUTH[2]	Base Services Target APF-Authorized Load Library	COMPWARE.MKMPnnn.SKMPAUTH
SKMPCNTL	Base Services Target Sample Library	COMPWARE.MKMPnnn.SKMPCNTL
SKMPLOAD[2]	Base Services Target Load Library	COMPWARE.MKMPnnn.SKMPLOAD
SMPMTS	Base Services SMP/E System File	COMPWARE.MKMPnnn.SMPMTS
SMPSCDS	Base Services SMP/E System File	COMPWARE.MKMPnnn.SMPSCDS
SMPSTS	Base Services SMP/E System File	COMPWARE.MKMPnnn.SMPSTS
ALHCCNTL	HCI Distribution Sample Library	COMPWARE.MLHCnnn.ALHCCNTL
ALHCLOAD	HCI Distribution Load Library	COMPWARE.MLHCnnn.ALHCLOAD
LHCnnnD	HCI Distribution Zone	COMPWARE.MLHCnnn.DZONE.CSI
LHCnnnT	HCI Target Zone	COMPWARE.MLHCnnn.TZONE.CSI
SLHCAUTH[2]	HCI Target APF-Authorized Load Library	COMPWARE.MLHCnnn.SLHCAUTH
SLHCCNTL	HCI Target Sample Library	COMPWARE.MLHCnnn.SLHCCNTL
SLHCLOAD[2]	HCI Target Load Library	COMPWARE.MLHCPnnn.SLHCLOAD
SMPMTS	HCI SMP/E System File	COMPWARE.MLHCnnn.SMPMTS
SMPSCDS	HCI SMP/E System File	COMPWARE.MLHCnnn.SMPSCDS
SMPSTS	HCI SMP/E System File	COMPWARE.MLHCnnn.SMPSTS

Table 2-3. Libraries Created During Installation With SMP/E

DDname	Library Type and Content	Dataset Name as Distributed
[1]These datasets are part of the Japanese Language support and are only installed if Japanese Language support is required. [2] For more information, see the *Abend-AID Installation and Customization Guide* or the *Abend-AID for CICS Installation and Customization Guide*.		
Note: The following dataset qualifiers are used in this table: • KCWnnn is a dummy FMID used to reserve space on the installation tape. • MLMSnnn specifies the LMS release, where nnn is the LMS release number. For example, MLMS101 indicates LMS 1.0.1. • LCXnnn specifies the CSS release, where nnn is the CSS release number. For example, LCX760 indicates CSS 7.6. • MKMPnnn specifies the Base Services release, where nnn is the release number. For example, MKMP520 indicates Base Services 5.2.0. • MLHCnnn specifies the HCI release. For example, MLHC250 specifies HCI 2.5.0.		

Note: KCWnnn is a dummy FMID used to reserve space on the installation tape for miscellaneous files used during installation and customization and for the maintenance files. This FMID (KCW001 for ECC, but may vary for other Compuware SMP/E-installed products) is not intended to be installed, but merely defines some reserved file space to SMP/E.

SMP/E Tape Layout

Refer to Table 2-4 for a list of the SMP/E files on the ECC distribution tape.

Table 2-4. ECC SMP/E Tape Layout

File Number	File Name on Tape	File Contents
1	CW.COMPWARE.SMPMCS	MCS Statements
2	COMPWARE.LKCW001.F1	SMP/E Installation Sample Library
3	COMPWARE.LKCW001.F2	PTFs/APARs for this release (cumulative maintenance)
4	COMPWARE.LKCW001.F3	HOLDDATA for maintenance
5	COMPWARE.LKCW001.F4	Dummy
6	COMPWARE.LKCW001.F5	PDS containing PTF abstracts
7	COMPWARE.MLCXnnn.F1	CSS JCLIN statements
8	COMPWARE.MLCXnnn.F2	CSS Object Modules
9	COMPWARE.MLCXnnn.F3	CSS Panels and SAMPENU members
10	COMPWARE.NLCXnnn.F1	CSS Japanese panels and messages
11	COMPWARE.MKMPnnn.F1	Base Services JCLIN
12	COMPWARE.MKMPnnn.F2	Base Services Object Modules
13	COMPWARE.MKMPnnn.F3	Base Services Panels
14	COMPWARE.OKMPnnn.F1	Base Services JCLIN for Japanese Language
15	COMPWARE.OKMPnnn.F2	Base Services Object Modules for Japanese Language
16	COMPWARE.MLMSnnn.F1	LMS JCLIN
17	COMPWARE.MLMSnnn.F2	LMS load library
18	COMPWARE.MLMSnnn.F3	LMS panels, skeletons, messages and CLIST
19	COMPWARE.MLHCnnn.F1	HCI JCLIN
20	COMPWARE.MLHCnnn.F2	HCI Object Modules

Table 2-4. ECC SMP/E Tape Layout

File Number	File Name on Tape	File Contents
21	COMPWARE.MLHCnnn.F3	HCI Panels and SAMPENU members
[1] NLCXnnn is the FMID for Japanese support for CSS. OKMPnnn is the FMID for Japanese support for Base Services.		

ECC Installation Considerations

This section describes considerations for installing and customizing ECC that applies to LMS, CSS, Base Services, and/or HCI.

Link Lists

If you place an ECC component load library in the link list, it must be placed ahead of all Compuware product libraries in the link list. Refer to your Compuware product documentation to determine whether the load library or libraries for your components should be placed in the link list.

PDS/E Support

ECC components support Partitioned Data Set/Extended (PDS/E) libraries. LMS, CSS, Base Services, and/or HCI can be installed in PDS/E object libraries. Input source members for the CSS preprocessor or compiled listings for the CSS postprocessor may reside in a data PDS/E library. The following are supported for PDS/E libraries:

- PDS/E object libraries are supported with DFSMS 1.1 and above.
- PDS/E data libraries are supported with DFSMS 1.0 and above.

CSS Installation Considerations

This section describes considerations for installing Compuware Shared Services.

APF Authorization

Following IBM system integrity guidelines, Compuware recommends that your CSS load library NOT be APF-authorized. Depending on your MVS system installation options, libraries placed in the link list may be APF-authorized automatically.

Note: If you are using Abend-AID and Abend-AID is APF-authorized, then CSS should also be APF-authorized. If Abend-AID is not APF-authorized, then CSS should not be APF-authorized. The ONLY time the CSS load library should be APF-authorized is when you are using the Abend-AID load library as APF-authorized.

CICS Startup JCL

The CSS library must be placed ahead of all Compuware product libraries in the DFHRPL concatenation of the CICS startup JCL for the following product:

- XPEDITER/CICS (all supported releases)

TSO Logon PROCs

If the CSS load library is not in the link list and dynamic library allocation is not used, the CSS load library must be placed before all Compuware product load libraries in the STEPLIB or ISPLLIB concatenations in the TSO logon PROC. The CSS message and panel

libraries must be placed before all Compuware product message and panel libraries in the ISPMLIB and ISPPLIB concatenations in the TSO logon PROC.

Beginning with CSS 6.0, the CSS load, message, and panel libraries can be dynamically allocated for use with Compuware/VF. See SLCXCNTL members CWVFCLSE, CWVFCLSJ, CWVFREXE, and CWVFREXJ.

Sharing CSS DDIO Files With Multiple CPUs

If you are sharing CSS DDIO files among multiple CPUs, the following qnames must be changed from LOCAL to GLOBAL enqueues.

- ABENDAID
- ABENDDAM
- ABENDSMF
- CWCATALG

Note: These qnames are needed for **all** products. Please refer to the individual Compuware product documents with which CSS was distributed for other product requirements.

For IBM's GRS (Global Resource Serialization), these qnames must be added to the inclusion RNL. This RNL is defined in the GRSRNLxx member of SYS1.PARMLIB for MVS/XA and later. For pre-MVS/XA systems, these qnames are maintained in the ISGSIRNL entry point of ISGGRNLO. Please consult the IBM product documentation for further details.

For resource serialization packages other than GRS (for example MIM), consult the product documentation for changing LOCAL to GLOBAL enqueues.

CSS Customer Modification Facility

The CSS Customer Modification Facility retrofits your site's CSS restricted zaps to the SMP/E environment. If you are migrating to the current CSS release from a CSS release prior to CSS 7.4, and have a restricted zaps applied to your system, you must use the CSS Customer Modification Facility. For more information on using this facility, refer to the "Using the Customer Modification Facility" appendix in the *Compuware Shared Services User/Reference Guide*.

CSS Language Processor (LP)

The language processor captures information about a compiler listing and stores it in a DDIO file. The majority of this information is gathered from the compiler listing. Optionally, SYSIN and SYSLIB are also examined.

The language processor can be run using two methods:

- preprocessor
- postprocessor

You may use the preprocessor or postprocessor (whichever is best for your site as needed.)

This section discusses the benefits of the preprocessor and the postprocessor and describes when to use them. See the appropriate language processor chapters in the Compuware Shared Services User/Reference Guide for additional information on determining when to use the pre- or postprocessor.

Preprocessor

Some information about a program module is not always available from the compiler listing. The preprocessor, in addition to gathering information from the compiler listing,

gathers information from SYSIN and SYSLIB. The preprocessor provides additional functionality for XPEDITER/CICS and XPEDITER/TSO, and is required for PL/I programs that use the %NOPRINT statement.

When setting up the language processors, Compuware recommends that you use the preprocessor instead of the postprocessor. The postprocessor should only be used in situations where compiled source listings have been stored.

Preprocessor Steps

1. Determines the proper compiler options required to process the listing file.
2. Automatically invokes the compiler or assembler to compile or assemble your source program.
3. Writes the listing to the DDIO file. An enhanced listing for COBOL programs can be produced, if desired. PL/I and Assembler listings are written to SYSPRINT. C listings are written to SYSCPRT.

Preprocessor Benefits

- **Better handling of compiler errors.** By using the preprocessor, you can avoid the potential problem of processing a listing with compiler errors. The preprocessor, in conjunction with the CONDDDIO parameter, internally checks the return code from the compiler and doesn't write a DDIO listing that contains errors. For example, if you set the CONDDDIO parameter to 8, the preprocessor will write the compiler listing to the DDIO file unless the compiler return code exceeds 8. The preprocessor can process the compiler errors more effectively than the postprocessor. Compuware recommends that you use the preprocessor rather than the postprocessor.
- **Capturing suppressed source code.** When any of the following parameters are used, sections of source code can be suppressed from the compiler listing:

For COBOL	COPY SUPPRESS
For PL/I	%NOPRINT
For Assembler	PRINT OFF or PRINT NOGEN
For C	NOSHOWINC and NOEXPMAC

  The preprocessor captures this information from SYSIN and SYSLIB, or by forcing on more revealing compiler options, then altering the listing to emulate the options you requested.
- **Automated compiler options.** The postprocessor requires that certain compiler options be specified in order to process all needed sections of the compiler listing. The preprocessor can automatically pass the required options to the compiler.
- **Simplified JCL.** While the postprocessor requires that the user add a step after the compile step, the preprocessor requires only that minor modifications be made to your existing compile step.

Files Dynamically Allocated

The following files are dynamically allocated by the Compuware preprocessor. The attributes can be overridden by specifying a DD in the JCL.

Table 2-5. Files Dynamically Allocated by the Preprocessor

DDname	Attributes
CWPWRKn (0-6)	BLKSIZE=19000 SPACE(TRK,(100,80)) UNIT=SYSDA
SORTWK01	BLKSIZE=19000 SPACE(TRK,(100,80)) UNIT=SYSDA
TEMPLIN	BLKSIZE=3200 SPACE(TRK,(200,100)) UNIT=SYSDA
CWPERRM	SYSOUT=*

Table 2-5. Files Dynamically Allocated by the Preprocessor

DDname	Attributes
SYSPRINT	SYSOUT=*
SYSOUT	SYSOUT=*
CWPPRTI (PL/I)	BLKSIZE=19000 SPACE(TRK,(100,200)) UNIT=SYSDA
CWPPRTI (all other languages)	BLKSIZE=16093 SPACE(TRK,(100,200)) UNIT=SYSDA DISP=MOD
CWPPDS2	SPACE (TRK,(100,80,80)) UNIT=SYSDA

CWPPRTO is dynamically allocated as a work file (for the preprocessor). SYSPRINT is where the listing goes.

Postprocessor

The postprocessor is executed as a single step and it can be a separate job. It reads in the listing created from the compiler. Information is gathered from the source listing, XREF, data maps, and object code sections of the listing. This information is stored in a DDIO file member and is used by various Compuware products.

While the preprocessor is the preferred method of loading the compiled listing to the DDIO, there are situations that necessitate the use of the postprocessor as a viable alternative. This is commonly found in production environments where the listings may be archived. In order for the listings to populate the DDIO, they must be compiled using the required options specified in the appropriate language processor chapters in the Compuware Shared Services User/Reference Guide. Note that these are usually the default options at many shops.

If the programs contain suppressed source code, certain additional control statements must be added to the language processor. Please view the Language Processor chapters in the Compuware Shared Services User/Reference Guide for the programming language desired for more details.

If the programs are compiled with OFFSET as well as OPT(imize), certain additional control statements must be added to the language processor. Please view the Language Processor chapters in the Compuware Shared Services User/Reference Guide for the programming language desired for more details.

Postprocessor Benefits

- Exact match of compiled output listings with the executable load module. This is especially important in production environments where time is critical.
- Eliminates the risk of copybook or source code changes prior to recompiling the code since the listing reflects the code that is being executed.
- Significantly less time is needed to process source listings as opposed to recompiling the source code.

Language Processor DD Statements

Table 2-6 lists the DD statements for the Assembler, COBOL, PL/I, and C language processors. Not every DD Name applies to every language processor.

Table 2-6. Language Processor DD Statements.

DDname	Purpose
CWPWRK (0-6)	LP work files
CWPPRMO	LP parameter input dataset
CWPDDIO	Target source listing DDIO file

Table 2-6. Language Processor DD Statements.

DDname	Purpose
CWPPRTI	Compiler listing input to postprocessor
CWPPRTO	Compiler listing output from postprocessor
CWPLOAD	Option LOAD or OBJECT module input to postprocessor
CWPDECK	Option DECK module input to postprocessor
CWPERRM	LP Error Messages

Note: CWPLOAD and CWPDECK correspond to the SYSLIN and SYSPUNCH compiler outputs. If you specify compiler option OBJECT, you should use CWPLOAD in your postprocessor JCL. If you specify compiler option DECK, you should use CWPDECK in your postprocessor JCL. Do not feed the SYSLIN output into CWPDECK or the SYSPUNCH output into CWPLOAD. The updated OBJECT contained in CWPLOAD and CWPDECK should be used in the program's linkedit step. The preprocessor does not use CWPLOAD and CWPDECK.

Writing a Listing to a DDIO File

Several options exist to place listings in a DDIO file. Each option requires that some changes be made to your JCL. The type of changes needed depend on whether you are using the preprocessor or the postprocessor, and the type of debugging session. Refer to "CSS Sample Library Members" for a list of the sample JCL that can be used for the various options.

Testing and Debugging Programs

The actual method used for handling testing and debugging programs depends on whether you are preprocessing or postprocessing:

- **Preprocessing** — Run the preprocessor and link your program.
- **Postprocessing** — Compile your program, run the postprocessor, and then link your program.

With either of these methods, the language processor places a listing in the DDIO file whenever a program is compiled.

Debugging Production Programs

If the original compiler listing is available in machine-readable format, you can retrieve the listing and use it as input to the postprocessor in order to reprocess the listing and recreate it in the DDIO file. The postprocessor can process listings stored by most products.

Note: If the listing you use as input to the postprocessor contains any other information (such as translator or link edit output), in addition to the compiler information, you may not obtain the desired results.

CSS Sample Library Members

The following list describes the members of the CSS sample JCL library that you may work with during installation (depending on your installation options). These members are distributed on the CSS distribution tape and updated via SMP/E. The members will be located in the following files after the Apply and Accept steps are complete.

- COMPWARE.MLCX*nnn*.SLCXCNTL (target library)

- COMPWARE.MLCX*nnn*.ALCXCNTL (distribution library)

where *nnn* indicates the CSS release number.

To print the CSS sample library members, submit the JCL contained in library member CXPRINT after the tape has been unloaded.

AACONVRT Sample JCL to convert existing Abend-AID report DDIO to shared directory report database

CWASSECD Security Exit program DSECT.

CWASSECU Sample Security Exit program.

CWCMTRHE Sample customized horizontal translation table for mixed-case English.

CWCMTRHT Sample customized horizontal translation table for uppercase English.

CWCMTRHU Sample customized horizontal translation table for mixed-case English with Euro character.

CWCMTRVE Sample customized vertical translation table for mixed-case English.

CWCMTRVT Sample customized vertical translation table for uppercase English.

CWCMTRVU Sample customized vertical translation table for mixed-case English with Euro character.

Note: The following five members are specific for Abend-AID. They allow browse access to diagnostic reports and source listings directly through ROSCOE or print access via a job submitted from ROSCOE to run in batch. Refer to the members for additional information.

CWROSBAT ROSCOE sample members.

CWROSBRW ROSCOE sample members.

CWROSCOE ROSCOE sample members.

CWROSDIR ROSCOE sample members.

CWROSPDF ROSCOE sample members.

CWUTCLSE Sample CLIST to invoke CSS online utilities with English messages and panels.

CWUTCLSJ Sample CLIST to invoke CSS online utilities with Japanese messages and panels.

CWUTREXE Sample REXX EXEC to invoke CSS online utilities with English messages and panels.

CWUTREXJ Sample REXX EXEC to invoke CSS online utilities with Japanese messages and panels.

CWVFCLPT CLIST to invoke the language processor debugging aid.

Note: The debugging aid is a diagnostic tool to be used only under the supervision of Compuware Technical Support in the event of a problem related to the language processor.

CWVFCLSE Sample CLIST to invoke Compuware/VF using English messages and panels.

CWVFCLSJ Sample CLIST to invoke Compuware/VF using Japanese messages and panels.

CWVFDB2 Sample CLIST to activate the File-AID DB2 interface to the Compuware Viewing Facility.

CWVFREXE Sample REXX EXEC to invoke Compuware/VF using English messages and panels.

CWVFREXJ Sample REXX EXEC to invoke Compuware/VF using Japanese messages and panels.

CWVFRXPT Sample REXX EXEC to invoke the Compuware language processor debugging aid.

Note: The debugging aid is a diagnostic tool to be used only under the supervision of Compuware Technical Support.

CXAADIRX Abend-AID CWAASDUT DIRX sample JCL.

CXAAEXPO Abend-AID CWAASDUT EXPORT command sample JCL.

CXAAIMPO Abend-AID CWAASDUT IMPORT command sample JCL.

CXAAMOVE Abend-AID CWAASDUT MOVE command sample JCL

CXAARPT Sample CWDDSUTL control statement for formatting report files.

CXALDDAA Sample VSAM parameters for allocating Abend-AID report files.

CXALDDSL Sample VSAM parameters for allocating source listing files.

CXALLBAA JCL for creating sequential Abend-AID database files.

CXALLBLP JCL for creating sequential source listing database files.

CXALLBSD JCL for creating Abend-AID for CICS sequential transaction databases.

CXALLDAA JCL for creating Abend-AID shared directories.

CXALLDLP JCL for creating Source Listing Shared Directories.

CXALLDS JCL to allocate a sequential DDIO file.

CXALLMC JCL for creating Abend-AID for CICS shared directories.

CXALLVAA JCL for creating VSAM Abend-AID database files.

CXALLVLP JCL for creating VSAM Source Listing Database files.

CXALLVS JCL to allocate a VSAM DDIO file.

CXALLVSD JCL for creating Abend-AID for CICS VSAM transaction databases.

CXASM JCL for running the Assembler language postprocessor.

CXASMPRE JCL for running the Assembler language preprocessor.

CXC Sample JCL for running the C postprocessor.

CXCAPTUR Sample JCL for creating a tape from the output of the CSS Problem Documentation Utility.

CXCFGEXT Sample JCL to extract a source file from the configuration module in a CSS load library.

CXCFGINI Sample source configuration file.

CXCFGSET Sample JCL to build a configuration module in a CSS load library, from a source configuration file you create.

CXCIREG Sample JCL to register the Contact Information dataset.

CXCIVSAM Sample JCL to create the Contact Information dataset.

CXCOBPRE JCL for running the COBOL language preprocessor.

CXCOB1 JCL for the step to be added after the compile step in your current COBOL compile and link edit JCL. This is used in order to process the compiler listing through the Compuware language postprocessor.

CXCOB2 JCL to process COBOL compiler listings stored in machine-readable format.

CXCOB99 Sample JCL for running the COBOL postprocessor.

CXCPRE Sample JCL for running the C preprocessor.

CXC2 Sample JCL to postprocess C compiler listings stored in machine-readable format.

CXDDSUTL Sample JCL for running the CWDDSUTL batch file utility.

CXDDUNLP Sample JCL to execute the CWDDUNLP source extraction utility.

CXEXPORT Sample JCL for running the CWDDSUTL EXPORT command.

CXFLAG Sample JCL for the CWDDSUTL utility FLAG command.

CXFMTDS JCL for formatting a DDIO file.

CXFMTLST Sample CWDDSUTL control statement for formatting a source listing file.

CXFXDIRX Abend-AID for CICS CWFXSDUT DIRX sample JCL.

CXFXEXPO Abend-AID for CICS CWFXSDUT EXPORT command sample JCL.

CXFXEXTL Sample JCL to export a Abend-AID for CICS transaction **and** a source listing to a single tape.

CXFXIMPO Abend-AID for CICS CWFXSDUT IMPORT command sample JCL.

CXFXMOVE Abend-AID for CICS CWFXSDUT MOVE command sample JCL.

CXIMPORT Sample JCL for running the CWDDSUTL IMPORT command.

CXJCLSEC JCL to assemble and link edit the Security Exit program.

CXJCLTRT JCL to assemble and link edit the custom translation tables.

CXLPASM Sample Assembler language processor options.

CXLPC	Sample C language processor options.
CXLPCOBB	Sample COBOL language processor options (batch programs).
CXLPCOBC	Sample COBOL language processor options (CICS programs).
CXLPDIRX	LP source CWDDLPUT DIRX sample JCL.
CXLPEXPO	LP source CWDDLPUT EXPORT command sample JCL.
CXLPIMPO	LP source CWDDLPUT IMPORT command sample JCL.
CXLPMOVE	LP source CWDDLPUT MOVE command sample JCL.
CXLPPLI	Sample PL/I language processor options.
CXLPTFS	Sample JCL to run SMPE LIST on CSS Target/Distribution zones.
CXLZAPS	Sample JCL to list the PTFs and APARs on a CSS load library
CXMODMAP	Sample JCL member to map the CSECTs in a load module.
CXOTFPRE	Sample CLIST stub for preprocessing exit for OTF
CXOTFPST	Sample CLIST stub for postprocessing exit for OTF
CXPDDSU	Sample JCL to print a listing or report from a DDIO file.
CXPLI	JCL for running the PL/I language postprocessor.
CXPLIPRE	JCL for running the PL/I language preprocessor.
CXPLI2	JCL to postprocess PL/I compiler listings stored in machine-readable format.
CXPRCTL	IEBPTPCH control statements for job CXPRINT.
CXPRINT	JCL to print the entire CSS installation library.
CXRELS	PTF/APAR tracking module. **Do not modify this member.**
CXR00003	Sample Tutorial main menu.
CXSPRE01	Sample Skeleton for on-the-fly preprocessing.
CXSPST01	Sample Skeleton for on-the-fly postprocessing.
CXTUTOR	Alternate Sample Tutorial main panel.
DDIODAYS	REXX utility to delete DDIO file members by age.
DDIODAYJ	Sample JCL for running the DDIODAYS REXX utility. This utility deletes DDIO file members by age.
FDXTRN01	File-AID DB2 Interface CLIST.
P@CPRE01	Sample panel for preprocessing exit for on the fly.
P@CPST01	Sample panel for postprocessing exit for on the fly.
P@HPRE01	Sample Help panel for preprocessing exit for on-the-fly processing.
P@HPST01	Sample Help panel for postprocessing exit for on-the-fly processing.

LMS Installation Considerations

This section describes considerations for installing and customizing LMS.

APF Authorization

Following IBM system integrity guidelines, Compuware requires that your LMS load library be APF-authorized. You must provide an APF-authorized load library into which you place the modules for LMSINIT since all load modules for LMSINIT must reside in an APF-authorized load library.

Execution Sequence

In order to ensure availability of Compuware products with LMS-controlled access, the LMS runtime environment must be established before any of those products are initialized. For this reason, Compuware strongly recommends that you institute an MVS start-up procedure based on the sample proc provided. It should run automatically as part of IPL and IML processing prior to any procs utilized by other Compuware products. It may be necessary to consult your site's MVS system programmer.

The License Management System Functions

The License Management System has three basic functions.

License Management Administration

You will create and maintain your License File using the License Administration Utility (LAU) installed with your License Management Software. Your License File is used as the source from which Compuware products will validate access for your site. When you receive License Certificates for Compuware products, your organization's License Administrator must use the LAU to **import** the License Certificates into the License File. Additional features of the LAU assist in the maintenance of your License Files. In addition to browsing the License Certificates in your License File, you may run a number of administrative reports from the LAU. These reports will reveal the contents of your License File, from a number of perspectives, allowing you to determine the state of the License Certificate for any particular Compuware product release.

Depending on the requirements of your organization, you may have more than one License File. Each License File can be centrally administered from the License Administration Utility.

Establishing the Runtime LMS Environment

You will use the License Files created and maintained by the License Administration Utility as input to a program that establishes your Compuware LMS runtime environment. This program is named LMSINIT. LMSINIT is the License Management System program that reads the License File[1] and constructs the License Cache and License Management System subsystem against which Compuware product runtime license access requests are later made.

Important: You must re-establish your runtime License Management System environment at each IPL, and whenever you have an update to your License File, the update must be made available to your MVS or OS/390 systems. At IPL startup, LMSINIT must be completed before starting up any Compuware product.

1. LMSINIT can read more than one License File if you are an organization that administers Compuware product access for more than one Compuware customer.

Validating Product Access Requests During Product Use

Your access to your Compuware products will be validated when you use your Compuware products. The Compuware product will make a request of the License Management System to determine if your site has a valid License Certificate for the product release. The requests are made by the product at various times during its execution depending on the product's needs. The Compuware product will access the LMS subsystem established, request License File information and act upon the information. If the License Certificate information is valid, users will proceed with their use of the product without disruption. Any abnormal License Management event detected will be reported to the product's user and may optionally be reported by e-mail to your organization's License Administrator. Optionally, these events may also be recorded in SMF Logging.

Base Services Installation Considerations

This section describes considerations for installing Base Services which is a requirement for Distributed Viewing Support (DVS) and Abend-AID for CICS.

APF Authorization

The Base Services component library (COMPWARE.MKMPnnn.SKMPAUTH) must be APF-authorized. This library is used in the server region for Abend-AID Distributed Viewing Support and Abend-AID for CICS (with or without DVS).

TSO Logon PROCs

If you are utilizing Abend-AID Distributed Viewing Support via the Compuware Viewing Facility, you should update the CWVFCLSE, CWVFCLSJ, CWVFREXE, and CWVFREXJ members to add the Base Services Viewing Support library to the ISPLLIB concatenations (COMPWARE.MKMPnnn.SKMPLOAD). It may also be added to your TSO logon PROC or added to the link list.

HCI Installation Considerations

This section describes considerations for installing the Host Communications Interface which is a requirement for Distributed Viewing Support and Abend-AID for CICS.

APF Authorization

The HCI component library (COMPWARE.MLHCnnn.SLHCAUTH) must be APF-authorized. This library is used in the server region for Abend-AID Distributed Viewing Support and Abend-AID for CICS (with or without DVS) as well as functions for other Compuware products. See Table 2-1 on page 2-1 for the specific products.

TSO Logon PROCs

If you are utilizing Abend-AID Distributed Viewing Support via the Compuware Viewing Facility, you should update the CWVFCLSE, CWVFCLSJ, CWVFREXE, and CWVFREXJ members to add the HCI library to the ISPLLIB concatenations (COMPWARE.MLHCnnn.SLHCLOAD). It may also be added to your TSO logon PROC or added to the link list.

Chapter 3.
Installing Enterprise Common Components

IMPORTANT:

If the ECC components exist on your system, you can use the enclosed tape as you would a product maintenance tape to update the components with the latest changes. However, any ECC components which have newer release numbers MUST be installed.

CAUTION:

To prevent possible conflicts, Compuware strongly recommends that all Compuware products and components distributed via SMP/E share a single Global zone reserved solely for Compuware products. Target and distribution zones should not be shared with other products or different releases of the same product.

This chapter explains the steps required to install the primary portions of Compuware's Enterprise Common Components (ECC), including:

- Compuware Shared Services (CSS)
- License Management System (LMS)
- Base Services
- Host Communications Interface (HCI)

The procedures in this chapter unload the ECC tape and install the components into datasets. Although the full ECC installation procedure is discussed in this chapter, you may only need to perform certain portions depending on which products are used at your site. You should already have determined in the previous chapter which components you need to install.

Except where otherwise noted, the return code from all jobs should be zero.

The ECC installation process utilizes System Modification Program Extended (SMP/E). For more information about this type of installation, and other installation considerations, please refer to **Chapter 2, "Preparing for ECC Installation"**. For more information about SMP/E, consult the *IBM SMP/E Reference* or the *IBM SMP/E User's Guide*.

Note: If you have already installed components from the ECC tape and you receive an additional ECC tape, that tape may be used as a maintenance tape. If you are installing a maintenance tape, run job member $RSTABS to obtain current PTF abstract information. Approved PTFs are normally added to the maintenance tape on a monthly basis. If you receive more than one tape dated within a given month, the maintenance on these tapes may be identical. See **Chapter 4, "Applying Maintenance"** for complete maintenance installation instructions.

Step 1. Unload the ECC Installation Sample Library

The ECC installation sample library contains JCL used to complete the ECC installation process. This library is contained on the second file of the ECC tape. After unloading the library, you can review member $$INDEX for a description of each of the supplied members.

1. Type the JCL shown in Figure 3-1, and modify it as follows:
   a. Replace the *tttttt* value for **VOL=SER=** with the installation tape volume serial number.
   b. Replace the *vvvvvv* value for **VOL=SER=** with a valid DASD volume serial number. If a DASD volser is not required at your site, omit the DASD specification.
   c. The **DSN=COMPWARE.KCW*nnn*.INSTALL** value should indicate the name of ***your*** SMP/E installation sample library. Update the DSN as necessary to follow your site's naming conventions.
2. Submit the JCL to unload the initial install library.

Figure 3-1. JCL to Unload SMP/E Installation Sample Library

```
//ECC1A    JOB..your job card here...
//UNLOAD   EXEC PGM=IEBCOPY
//SYSUT3   DD UNIT=SYSDA,SPACE=(TRK,5)
//SYSUT4   DD UNIT=SYSDA,SPACE=(TRK,5)
//SYSPRINT DD SYSOUT=*
//TAPE     DD DSN=COMPWARE.LKCW001.F1,
//            UNIT=CART,                          <==VERIFY
//            DISP=(SHR,KEEP),
//            LABEL=(2,SL),
//            VOL=SER=tttttt                      <==VERIFY
//INSTLIB  DD DSN=COMPWARE.KCWnnn.INSTALL,        <==VERIFY
//            UNIT=SYSDA,
//            VOL=SER=vvvvvv,                     <==VERIFY
//            SPACE=(TRK,(45,15,20)),
//            DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB),
//            DISP=(,CATLG)
//SYSIN    DD *
  COPY  INDD=TAPE,OUTDD=INSTLIB
/*
```

Step 2. Select Installation Components

The ECC Installation JCL Customization Facility uses panels to gather your installation information. It then uses that information to build the jobs needed to perform the SMP/E installation.

Note: The Installation JCL Customization Facility requires a minimum of ISPF/PDF version 3.5. REXX is also required.

For information on resolving error messages from the installation facility, refer to the *Enterprise Common Components Messages and Codes* guide.

Begin the process of installing LMS, CSS, Base Services and/or HCI using the Installation JCL Customization Facility as follows:

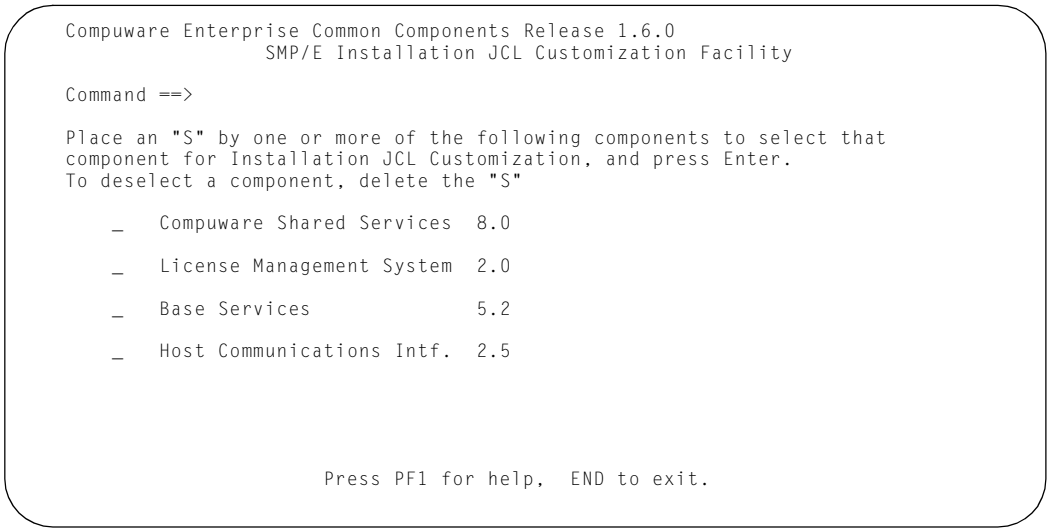
1. Invoke the installation facility by entering the TSO EXECUTE command with the dataset name of the installation library followed by ECCINST. For example:

   ```
   TSO EX 'COMPWARE.KCWnnn.INSTALL(ECCINST)'
   ```

   where COMPWARE.KCWnnn.INSTALL is the name of your installation sample library as you entered it in the JCL in the previous step.

   The entry panel for the Installation JCL Customization Facility (Figure 3-2) will be displayed.

Figure 3-2. Selecting Components to Install



2. Enter an **S** to the left of each component you want to install. Remember the following:
   - If you only require CSS, select only CSS on this panel.
   - If you only require LMS, select only LMS on this panel.
   - If you use Abend-AID 9.0 or more current and plan to use Distributed Viewing Support, you must install the CSS, Base Services, and HCI components.
   - If you use Abend-AID for CICS, you must install all four components.

   Note: If you are installing more than one component, Compuware recommends installing them simultaneously.

To indicate your choices, the word **SELECTED** will be displayed next to each component chosen. Figure 3-3 shows the confirmation text that would be displayed if CSS, LMS, Base Services, and HCI were all selected for installation.

Figure 3-3. Confirming Installation Selections

```
              Compuware Enterprise Common Components Release 1.6.0
                 SMP/E Installation JCL Customization Facility

Command ==>

Place an "S" by one or more of the following components to select that
component for Installation JCL Customization.
To deselect a component, delete the "S"

    S   Compuware Shared Services       8.0      SELECTED

    S   License Management System       2.0      SELECTED

    S   Base Services                   5.2      SELECTED

    S   Host Communications Interface   2.5      SELECTED


Press Enter to begin JCL Customization for the selected component(s).
Select all desired components before continuing.

                      Press PF1 for help,  END to exit.
```

3. To modify a selection, add or delete the **S** to the left of the component.
4. When you are satisfied with your selections, press Enter again to start the JCL customization process.

Step 3. Specify ECC Installation Environment Information

The panel shown in Figure 3-4 is displayed when you pressed Enter to confirm your component selections in the previous step.

Figure 3-4. Installation Environment Panel

```
                        Enterprise Common Components
                  SMP/E Installation JCL Customization Facility
Command ==>
Customizing: Compuware Shared Services       License Management
             Base Services                   Host Communications Intf.

Please enter the following parameters.

Do you want to enter SMS parameters?        NO
DASD Unit:                                  SYSDA
DASD Volser (optional):
SMP/E Work Unit:                            VIO
Installation Tape Volser:                   CN0000
Tape unit:                                  3490
Does your installation use CA-TMS?          NO
GLOBAL SMP/E dataset High-level Qualifier:  COMPWARE.GLOBAL


  Press Enter to continue, PF1 for help, or END to return to previous panel.
```

This panel prompts you to enter information about the installation environment. Notice that the names of the components selected for installation are listed below the Command prompt. Those names will remain there during the rest of this procedure.

1. Specify the installation tape volume serial number, device, and tape unit for the ECC tape as follows:

   SMS Parameters

   If you specify YES in the SMS Parameters field, you will be prompted to enter SMS (Storage Management Subsystem) parameters on the next panel. Depending on your site's SMS installation options, default SMS parameters may be supplied automatically. In this case, you may not need to specify SMS parameters to the Installation JCL Customization Facility. If you want to supply SMS parameters, specify YES in this field. Otherwise specify NO. The default value is NO.

   DASD Unit

   Enter a valid DASD unit name. The value of this field will be used in the UNIT= parameter of the DD statements that allocate non-VSAM libraries and datasets. The default value is SYSDA.

   DASD Volser

   This field is optional. If a volume is specified, SMP/E will use it as the target for library and dataset allocation.

   SMP/E Work Unit

   Enter the name of a DASD unit to use for allocating SMP/E work files. The default value is VIO.

   Installation Tape Volser

   Enter the serial number of the ECC tape.

Tape Unit

Enter the value for the UNIT= parameter of the tape input DD statements. This is a site-dependent parameter. CART, T3490, TAPE, etc. are examples of typical unit parameters. Enter an appropriate value based on your site's standards. The default value is 3490.

CA-TMS

Enter YES or NO in the CA-TMS field. CA-TMS sometimes requires a special parameter (EXPDT=98000) on the DD statement to process labeled tapes created at another site. Specify YES to add this value to the generated JCL members. If your site does not use CA-TMS or does not require this parameter, specify NO. The default value is NO.

Global SMP/E dataset High-Level Qualifier

Enter a value to be used for generating the names of the SMP/E system datasets. These datasets include SMPLOG, SMPPTS, SMPTLIB, and the Global Consolidated Software Inventory (CSI).

Note: Due to limits on the length of the DSPREFIX parameter (used to build the SMPTLIB names for SMP/E RELFILES), this field cannot be longer than 26 characters.

2. When you have completed the fields as required, press Enter to continue.

Step 4. Specify SMS Parameters

If, in the previous step, you did not specify that you wanted to enter SMS parameters, you can skip this step.

If you specified, on the Installation Environment Panel (Figure 3-4 on page 3-5), that you wanted to enter SMS parameters, the panel shown in Figure 3-5 is displayed when you press Enter.

Figure 3-5. Specifying SMS Parameters

```
                        Enterprise Common Components
                  SMP/E Installation JCL Customization Facility
Command ==>
Customizing: Compuware Shared Services       License Management
             Base Services                   Host Communications Intf.

Enter any or all of the following SMS parameters.

Data Class:        dataclas

Storage Class:     storclas

Management Class:  mgmtclas




  Press Enter to continue, PF1 for help, or END to return to previous panel.
```

1. Specify the data class, storage class, and/or management class parameters for the SMS operating environment as follows:

   Data Class

   Specify the eight-character data class name for SMS.

   Storage Class

   Specify the eight-character storage class name for SMS.

   Management Class

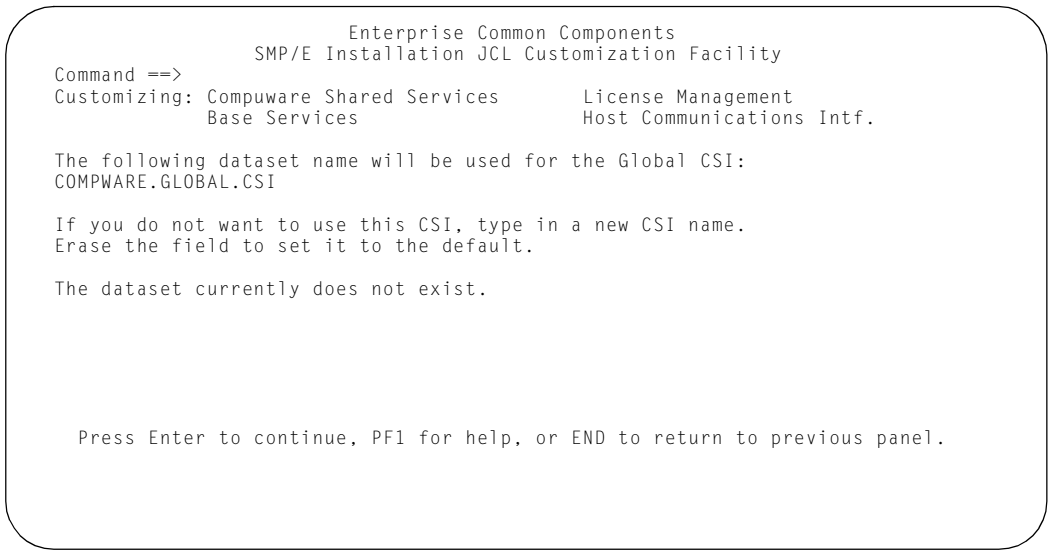
   Specify the eight-character management class name for SMS.

2. When you complete the fields as required, press Enter to continue.

Step 5. Specify Global CSI Dataset Name

The panel shown in Figure 3-6 is displayed when you pressed Enter in the previous step.

Figure 3-6. Verifying Global CSI Dataset Name



Note: The DSN for a new global CSI cannot exceed 38 characters. If the global CSI is predefined outside of the Customization Facility, it can consist of up to 44 characters.

1. Verify that the global CSI dataset name is correct. Change the name or specify a new name if necessary.
2. Press Enter.

   The Installation JCL Customization Facility determines whether the dataset currently exists. One of the following will occur:

If:	**Then:**
The dataset exists but has not been set up for use with Compuware products.	A jobstream is generated to add the appropriate parameters to the CSI. This job is named $*xx*I0GBA, where *xx* specifies the components you are installing: • **CS** for CSS only • **LM** for LMS only • **DV** for Base Services only • **HC** for HCI only • **EC** for two or more of the above components Review the resulting jobstream and compare the generated parameters to any existing SMP/E parameters.
The dataset exists and is already set up for use with Compuware products.	Job $*xx*I0GBA is not created.
The dataset does not exist.	Jobstream $xxI0GBA is generated to build the dataset and initialize it with the required Compuware SMP/E parameters.

Step 6. Specify Assembler and Jobcard Information

The panel shown in Figure 3-7 is displayed when you pressed Enter in the previous step. It prompts you for assembler and job card information.

Figure 3-7. Entering Assembler and Job Card Information

```
                        Enterprise Common Components
                  SMP/E Installation JCL Customization Facility
Command ==>
Customizing: Compuware Shared Services       License Management
             Base Services                   Host Communications Intf.

Default Assembler: ASMA90

Enter Job Card Information:
  //$$$$$$$$ JOB ('ACCT'),'USER NAME',
  //         CLASS=?,MSGCLASS=?,NOTIFY=PROGRAMMER
  //*        OPTIONAL JES PARAMETER CARD COULD GO HERE
  //*
  //*

NOTE: The class selected MUST allow enough CPU time for long running jobs.
      Jobs $xxI2APL and $xxI3ACC can easily exceed 5 CPU minutes each.


  Press Enter to continue, PF1 for help, or END to return to previous panel.
```

1. Specify the default assembler and job card information for the Installation JCL Customization Facility as follows:

   Default Assembler

   Some Compuware product installations require the assembly of one or more modules. Enter the program name of an assembler that resides in the linklist. The default assembler in the linklist for OS/390 1.2 and more current is ASMA90. The default prior to OS/390 1.2 is IEV90.

   Job Card Information

   Update the supplied model job card to follow your site standards. You may enter up to five lines of job information.

   By default, the installation facility will set the *$$$$$$$$* value equal to the member name of each job created. You also have the option of specifying the job name using the format *xxxxxxx$* or *xxxxxxxx*, where the *x*'s are whatever name you choose. If you set it to *xxxxxxx$*, the installation facility will set the job name for each job to *xxxxxxx* and append a number to it. A letter will be appended for jobs 10 and above. If you set it to *xxxxxxxx*, the installation facility will use the specified name for all jobs created.

   Note: The job class specified **must** allow enough CPU time for long-running jobs.

2. After you have supplied the required information, press Enter to continue.

Step 7. Specify SMP/E High-Level Qualifiers

The panel shown in Figure 3-8 is displayed when you pressed Enter in the previous step. It prompts you for the SMP/E high-level qualifiers of the components previously selected for installation.

Figure 3-8. Specifying SMP/E High-Level Qualifiers

```
                        Enterprise Common Components
                  SMP/E Installation JCL Customization Facility
Command ==>
Customizing: Compuware Shared Services       License Management
             Base Services                   Host Communications Intf.

Enter the following information:
  Compuware Shared Services
  SMP/E Dataset High-level Qualifier: COMPWARE.MLCXnnn

  License Management
  SMP/E Dataset High-level Qualifier: COMPWARE.MLMSnnn

  Base Services
  SMP/E Dataset High-level Qualifier: COMPWARE.MKMPnnn

  Host Communications Intf.
  SMP/E Dataset High-level Qualifier: COMPWARE.MLHCnnn


  Press Enter to continue, PF1 for help, or END to return to previous panel.
```

Note: If a high-level qualifier is longer than 28 characters, the installation facility is not able to generate a default dataset name for the target and distribution zones. You can either change the high-level qualifier entered, or manually modify the zone dataset names to valid lengths.

1. Specify the CSS, LMS, Base Services, and/or HCI high-level qualifiers as applicable for your installation as follows:

   For each component, enter a value to be used in generating names of the specific datasets, including target and distribution zone CSIs, and their respective non-SMP/E target and distribution libraries. The value entered cannot exceed 35 characters.

2. After you supply the appropriate high-level qualifiers, press Enter. You will be prompted for the CSS, LMS, Base Services, and/or HCI target and distribution zones, depending on which components you are installing.

3. Continue with the appropriate steps below, skipping any that do not apply.

Step 8. Specify CSS Target and Distribution Zones

If you are not installing CSS, you may skip this step.

If you are installing the CSS component, you will be prompted to specify the CSS target and distribution zones as shown in Figure 3-9.

Figure 3-9. Specifying CSS Target and Distribution Zones

```
                        Enterprise Common Components
                  SMP/E Installation JCL Customization Facility
Command ==>
Customizing: Compuware Shared Services       License Management
             Base Services                   Host Communications Intf.

Enter the following information for Compuware Shared Services

Distribution Zone Name:            LCXnnnD
DZone CSI Dataset Name:            COMPWARE.MLCXnnn.DZONE.CSI
DZone CSI currently does not exist


Target Zone Name:                  LCXnnnT
TZone CSI Dataset Name:            COMPWARE.MLCXnnn.TZONE.CSI
TZone CSI currently does not exist

Japanese Language Support?         NO

Warning: Dzone and/or Tzone name differ from default for CSS release n.n
Zone CSI exists, has Zone Name already been defined?     NO

  Press Enter to continue, PF1 for help, or END to return to previous panel.
```

1. Specify the CSS target and distribution zone information as follows:

   Distribution Zone Name

   Specifies the name of the SMP/E distribution zone used by CSS. The name you specify must be unique within the global zone. The default is LCXnnnD.

   DZone CSI Dataset Name

   Specifies the dataset name of the SMP/E distribution zone used by CSS. The default is COMPWARE.MLCXnnn.DZONE.CSI.

   Distribution CSI Currently exists/does not exist

   Indicates whether the CSS distribution zone CSI dataset exists.

   Target Zone Name

   Specifies the name of the SMP/E target zone used by CSS. The name you specify must be unique within the global zone. The default is LCXnnnT.

   Target Zone CSI Dataset Name

   Specifies the dataset name of the SMP/E target zone used by CSS. The default is COMPWARE.MLCXnnn.TZONE.CSI.

   Target CSI Currently exists/does not exist

   Indicates whether the CSS target zone CSI dataset exists.

   Japanese Language Support

   CSS can allocate message and panel libraries to display Japanese language characters rather than English. Specify YES to allocate and install the Japanese language message and panel libraries. Specify NO to bypass allocating these libraries.

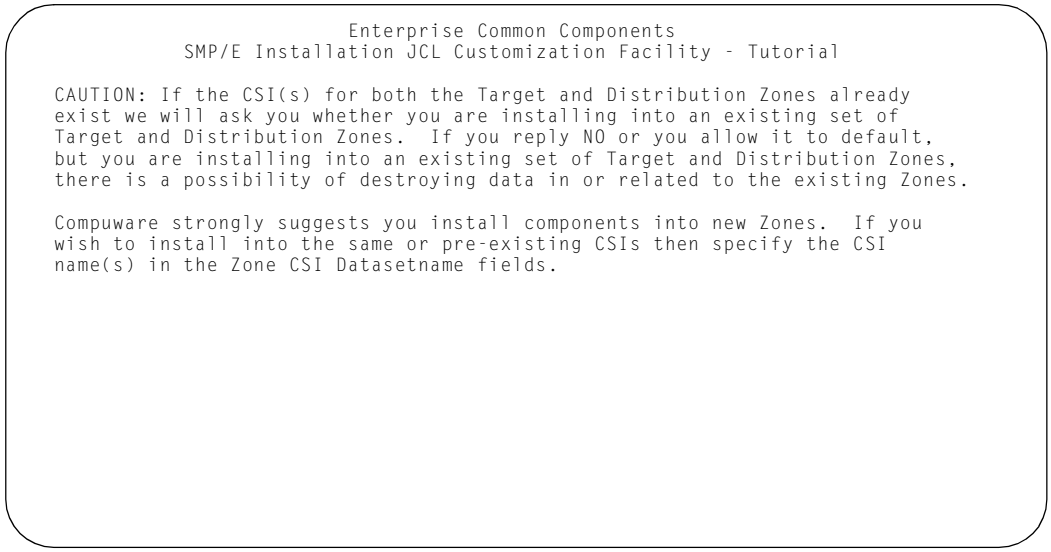
Warning: Dzone and/or Tzone name differ from default for CSS release n.n

You will get this warning message only if the distribution and/or target zone names entered are different from the default.

Zone CSI exists, has Zone Name already been defined?

You will get this Zone message if the CSIs for both the Target and Distribution Zones already exist. See Figure 3-10 for an explanation.

Figure 3-10. Installing into Pre-existing Target/Distribution Zones



Note: You will only get the above screen if the Zone CSI exists message appears on Figure 3-9 and you reply NO.

2. After you supply the appropriate information, press Enter. You may be prompted for the LMS, Base Services, and/or HCI target and distribution zones, depending on which components you are installing.
3. Continue with the next appropriate steps, skipping any that do not apply.

Step 9. Specify LMS Target and Distribution Zones

If you are not installing LMS, you may skip this step.

If you are installing LMS, you will be prompted to specify the LMS target and distribution zones as shown in Figure 3-11.

Figure 3-11. Specifying the LMS Target and Distribution Zones

```
                        Enterprise Common Components
                  SMP/E Installation JCL Customization Facility
Command ==>
Customizing: Compuware Shared Services       License Management
             Base Services                   Host Communications Intf.

Enter the following information for License Management

Distribution Zone Name:            LMSnnnD
DZone CSI Dataset Name:            COMPWARE.MLMSnnn.DZONE.CSI
DZone CSI currently does not exist


Target Zone Name:                  LMSnnnT
TZone CSI Dataset Name:            COMPWARE.MLMSnnn.TZONE.CSI
TZone CSI currently does not exist


Warning: Dzone and/or Tzone name differ from default for LMS release 2.0
Zone CSI exists, has Zone Name already been defined?     NO



  Press Enter to continue, PF1 for help, or END to return to previous panel.
```

1. Specify the LMS target and distribution zone information as follows:

   Distribution Zone Name

   Specifies the name of the SMP/E distribution zone used by LMS. The name you specify must be unique within the global zone. The default is LMSnnnD.

   DZone CSI Dataset Name

   Specifies the dataset name of the SMP/E distribution zone used by LMS. The default is COMPWARE.MLMSnnn.DZONE.CSI.

   Distribution CSI Currently exists/does not exist

   Indicates whether the distribution zone CSI dataset exists.

   Target Zone Name

   Specifies the name of the SMP/E target zone used by LMS. The name you specify must be unique within the global zone. The default is LMSnnnT.

   Target Zone CSI Dataset Name

   Specifies the dataset name of the SMP/E target zone used by LMS. The default is COMPWARE.MLMSnnn.TZONE.CSI.

   Target CSI Currently exists/does not exist

   Indicates whether the target zone CSI dataset exists.

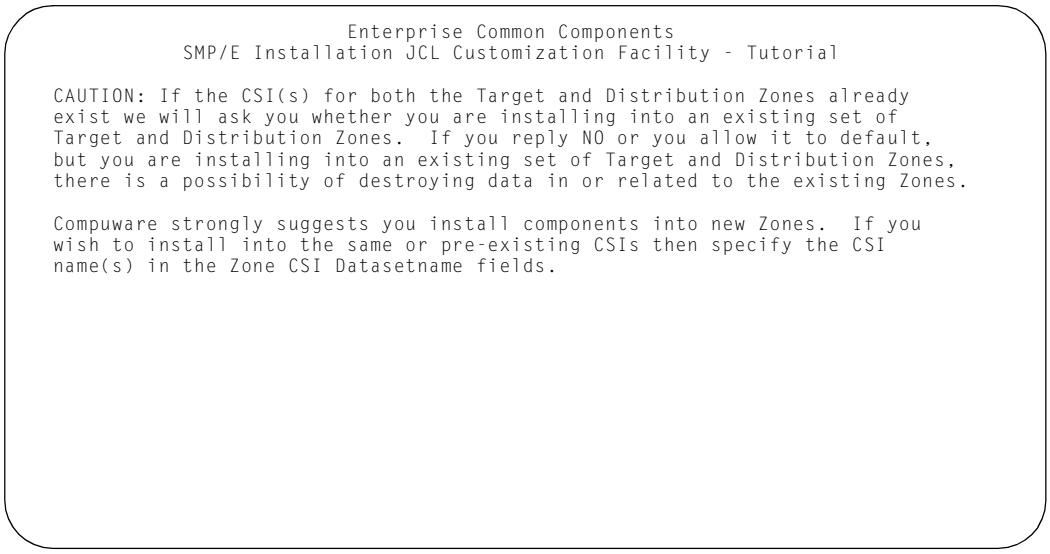
   Warning: Dzone and/or Tzone name differ from default for LMS release 2.0

   You will get this warning message only if the distribution and/or target zone names entered are different from the default.

Zone CSI exists, has Zone Name already been defined?

You will get this Zone message if the CSIs for both the Target and Distribution Zones already exist. See Figure 3-12 for an explanation.

Figure 3-12. Installing into Pre-existing Target/Distribution Zones



Note: You will only get the above screen if the Zone CSI exists message appears on Figure 3-11 and you reply **NO**.

2. After you supply the appropriate information, press Enter.

Step 10. Specify Language Environment Link Library for LMS

If you are not installing LMS, you may skip this step.

If you are installing LMS, you will be prompted to specify the IBM Language Environment (LE) link library, as shown in Figure 3-13.

Figure 3-13. Specifying the Language Environment Link Library for LMS

```
                          Enterprise Common Components
                    SMP/E Installation JCL Customization Facility
 Command ==>
Customizing: License Management


Enter the following information for License Management


 Language Environment (LE) link library:
 CEE.SCEELKED

 The library currently does exist


 This library is required to install and maintain LMS modules.
 A minimum of LE version 1.5 is required.



   Press Enter to continue, PF1 for help, or END to return to previous panel.
```

1. Specify the IBM Language Environment link library as follows:

   Language Environment (LE) link library

   Beginning with LMS Release 2.0, you must specify your site's LE link library for the proper installation and maintenance of LMS.

   After you enter the LE link library dataset name, this screen displays whether the library exists.

2. After you enter the library name and verify its existence, press Enter.
3. Continue with the next step, or skip it if it does not apply.

Step 11. Specify Base Services Target and Distribution Zones

If you are not installing Base Services, you may skip this step. Base Services is required if you are installing Abend-AID for CICS or Abend-AID with Distributed Viewing support.

If you are installing the Base Services component, you will be prompted to specify the target and distribution zones as shown in Figure 3-14.

Figure 3-14. Specifying Base Services Target and Distribution Zones

```
                        Enterprise Common Components
                  SMP/E Installation JCL Customization Facility
Command ==>
Customizing: Compuware Shared Services       License Management
             Base Services                   Host Communications Intf.

Enter the following information for Base Services

Distribution Zone Name:            KMPnnnD
DZone CSI Dataset Name:            COMPWARE.MKMPnnn.DZONE.CSI
DZone CSI currently does not exist


Target Zone Name:                  KMPnnnT
TZone CSI Dataset Name:            COMPWARE.MKMPnnn.TZONE.CSI
TZone CSI currently does not exist

Japanese Language Support?         NO

Warning: Dzone and/or Tzone name differ from default for DVS release 5.2
Zone CSI exists, has Zone Name already been defined?     NO

  Press Enter to continue, PF1 for help, or END to return to previous panel.
```

1. Specify Base Services target and distribution zone information as follows:

   Distribution Zone Name

   Specifies the name of the SMP/E distribution zone used by Base Services. The name you specify must be unique within the global zone. The default is KMPnnnD.

   DZone CSI Dataset Name

   Specifies the dataset name of the SMP/E distribution zone used by Base Services. The default is COMPWARE.MKMPnnn.DZONE.CSI.

   Distribution CSI Currently exists/does not exist

   Indicates whether the Base Services distribution zone CSI dataset exists.

   Target Zone Name

   Specifies the name of the SMP/E target zone used by Base Services. The name you specify must be unique within the global zone. The default is KMPnnnT.

   Target Zone CSI Dataset Name

   Specifies the dataset name of the SMP/E target zone used by Base Services. The default is COMPWARE.MKMPnnn.TZONE.CSI.

   Target CSI Currently exists/does not exist

   Indicates whether the target zone CSI dataset exists.

Japanese Language Support

Abend-AID for CICS and Abend-AID with Distributed Viewing Support can display Japanese language characters rather than English. Specify YES to install Japanese language support. Specify NO to bypass installing this support.

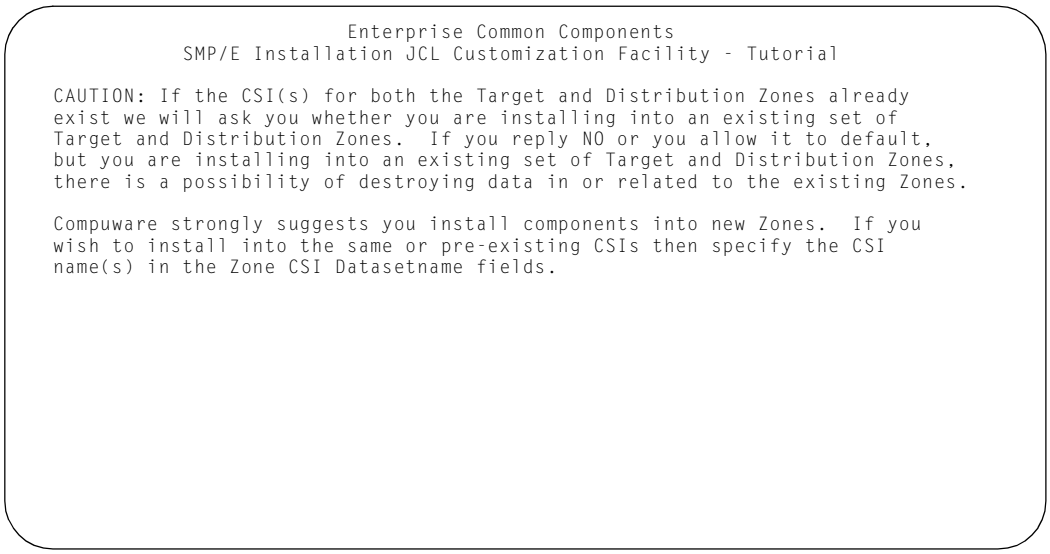
Warning: Dzone and/or Tzone name differ from default for Base Services release 5.2.

You will get this warning message only if the distribution and/or target zone names entered are different from the default.

Zone CSI exists, has Zone Name already been defined?

You will get this Zone message if the CSIs for both the Target and Distribution Zones already exist. See Figure 3-15 for an explanation.

Figure 3-15. Installing into Pre-existing Target/Distribution Zones



Note: You will only get the above screen if the Zone CSI exists message appears on Figure 3-14 and you reply NO.

2. After you supply the appropriate information, press Enter. You may be prompted for the LMS target and distribution zones, depending on which components you are installing.
3. Continue with the next step, or skip it if it does not apply.

Step 12. Specify HCI Target and Distribution Zones

If you are not installing the Host Communications Interface, you may skip this step. HCI is required if you are installing Abend-AID for CICS or Abend-AID with Distributed Viewing support.

If you are installing the HCI component, you will be prompted to specify the target and distribution zones as shown in Figure 3-16.

Figure 3-16. Specifying HCI Target and Distribution Zones

```
                        Enterprise Common Components
                  SMP/E Installation JCL Customization Facility
Command ==>
Customizing: Compuware Shared Services       License Management
             Base Services                   Host Communications Intf.

Enter the following information for Host Communications Interface

Distribution Zone Name:            LHCnnnD
DZone CSI Dataset Name:            COMPWARE.MLHC250.DZONE.CSI
DZone CSI currently does not exist


Target Zone Name:                  LHCnnnT
TZone CSI Dataset Name:            COMPWARE.MLHC250.TZONE.CSI
TZone CSI currently does not exist

Japanese Language Support?         NO

Warning: Dzone and/or Tzone name differ from default for HCI release 2.5
Zone CSI exists, has Zone Name already been defined?     NO

  Press Enter to continue, PF1 for help, or END to return to previous panel.
```

1. Specify HCI target and distribution zone information as follows:

   Distribution Zone Name

   Specifies the name of the SMP/E distribution zone used by HCI. The name you specify must be unique within the global zone. The default is KMPnnnD.

   DZone CSI Dataset Name

   Specifies the dataset name of the SMP/E distribution zone used by HCI. The default is COMPWARE.MKMPnnn.DZONE.CSI.

   Distribution CSI Currently exists/does not exist

   Indicates whether the HCI distribution zone CSI dataset exists.

   Target Zone Name

   Specifies the name of the SMP/E target zone used by HCI. The name you specify must be unique within the global zone. The default is KMPnnnT.

   Target Zone CSI Dataset Name

   Specifies the dataset name of the SMP/E target zone used by HCI. The default is COMPWARE.MKMPnnn.TZONE.CSI.

   Target CSI Currently exists/does not exist

   Indicates whether the target zone CSI dataset exists.

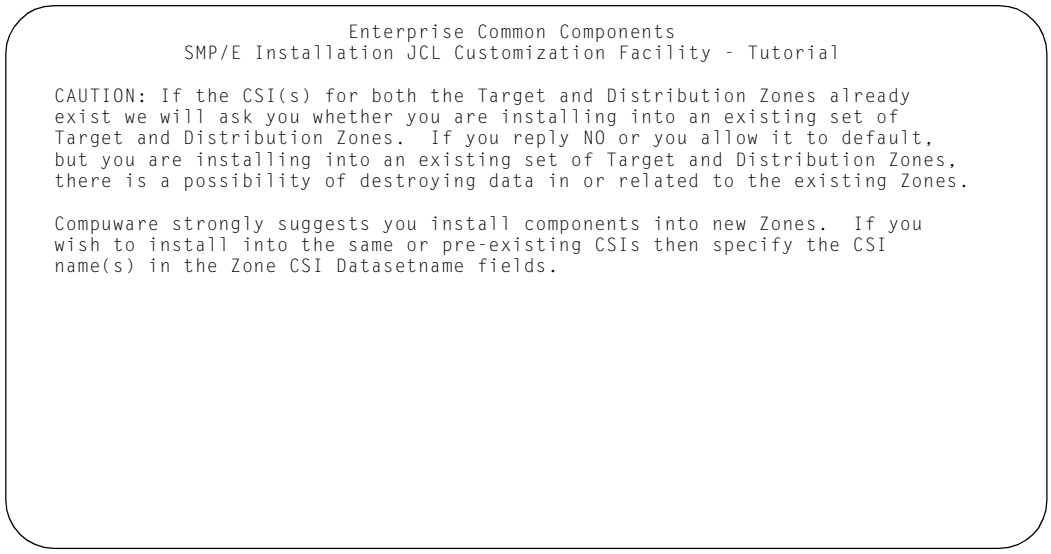
   Warning: Dzone and/or Tzone name differ from default for HCI release 2.5.

   You will get this warning message only if the distribution and/or target zone names entered are different from the default.

Zone CSI exists, has Zone Name already been defined?

You will get this Zone message if the CSIs for both the Target and Distribution Zones already exist. See Figure 3-17 for an explanation.

Figure 3-17. Installing into Pre-existing Target/Distribution Zones



Note: You will only get the above screen if the Zone CSI exists message appears on Figure 3-16 and you reply NO.

2. After you supply the appropriate information, press Enter.
3. Continue with the next step, verification of the information that was just entered.

Step 13. Verify Installation Information

When you pressed Enter in the previous step, the first of a series of confirmation panels was displayed. These panels verify all of the information you have entered up to this point.

- To accept the information on each confirmation panel, press Enter.
- To correct a particular parameter, press END to return to the associated entry panel.

After you accept the information on each of the confirmation panels, the installation process will continue.

Step 14. Process the JCL

The panel shown in Figure 3-18 is displayed when you pressed Enter the final time in the previous step.

Figure 3-18. Confirming JCL Processing

```
                        Enterprise Common Components
                  SMP/E Installation JCL Customization Facility

                             Processing Verification Screen
Command ==>
Customizing: Compuware Shared Services       License Management
             Base Services

Press Enter to begin customizing the JCL.
```

1. Press Enter to start the processing that builds customized JCL members. Press END to cancel the installation.

   When you press Enter, the progress of the JCL customization process is displayed on the screen. After the installation JCL has been created, a panel is displayed informing you that a parameter report has been generated (Figure 3-19). This report details the JCL created and lists the parameters and execution information used during installation.

Figure 3-19. Report Member Detailing JCL Processing

```
                        Enterprise Common Components
                  SMP/E Installation JCL Customization Facility

The following PDS member contains a report of all parameters used during
customization and a log of all JCL members created:


COMPWARE.KCWnnn.INSTALL(xxREPORT)


                  Press Enter to browse the report, END to exit.
```

2. If you want to browse the report at this time, press Enter. You can also examine the report later by accessing the applicable installation library report member shown on this panel:
   - CSREPORT (CSS only)
   - LMREPORT (LMS only)
   - DVREPORT (Base Services only)
   - HCREPORT (HCI only)
   - ECREPORT (Two or more components)
3. The processing of the Installation JCL Customization Facility is now finished. Press END to exit.

Jobs Created by the Installation JCL Customization Facility

All of the jobs created during installation customization are described below. Note that in addition to component-specific jobs, a series of jobs used during the maintenance cycle are also created. See “Maintenance Jobs” on page 3-23 for more information.

Jobs Common to All Components

The jobs in this section are customized for use when you install an ECC component. The *xx* that appears in the job names is replaced with a two-letter code during installation. Which two-letter code is used depends on which components you are installing:

- **CS** (CSS only)
- **LM** (LMS only)
- **DV** (Base Services only)
- **HC** (HCI only)
- **EC** (Two or more components)

$xxI0GBA Allocates and/or initializes the Compuware Global CSI and Global SMP/E datasets.

Note: This job is not built if the Global CSI already exists and you indicate it is already set up for Compuware use.

$xxI1PRA Allocates the CSS, LMS, Base Services and/or HCI libraries and SMP/E datasets.

$xxI1PRB Initializes the CSS, LMS, Base Services and/or HCI SMP/E target and distribution zone CSIs. It then adds the required CSS, LMS, Base Services, and/or HCI entries to the Global Zone CSI.

$xxI2RCV Receives the CSS, LMS, Base Services and/or HCI base code and maintenance.

$xxI3APL Applies the CSS, LMS, Base Services and/or HCI base code and maintenance.

$xxI4ACC Accepts the CSS, LMS, Base Services and/or HCI base code and maintenance.

$PRTABS Optional job to print abstracts of all PTFs in the maintenance file.

$RSTABS JCL to unload current PTF abstracts from an installation or maintenance tape.

CSS Installation Jobs

The following jobs are customized for installing CSS only.

$CSPABS Optional job that lists current CSS PTF abstracts.

$CSLPTFS Lists all PTFs applied to the CSS target and distribution zones. This optional job is for reporting purposes only.

LMS Installation Jobs

The following jobs are customized for installing LMS only.

$LMPABS Optional job that lists current LMS PTF abstracts.

$LMLPTFS Lists all PTFs applied to LMS target and distribution zones. This optional job is for reporting purposes only.

Base Services Installation Jobs

The following jobs are customized for installing Base Services only.

$DVPABS Optional job that lists current Base Services PTF abstracts.

$DVLPTFS Lists all PTFs applied to the Base Services target and distribution zones. This optional job is for reporting purposes only.

HCI Installation Jobs

The following jobs are customized for installing HCI only.

$HCPABS Optional job that lists current HCI PTF abstracts.

$HCLPTFS Lists all PTFs applied to the HCI target and distribution zones. This optional job is for reporting purposes only.

Maintenance Jobs

The following jobs may also be built by the installation facility but are used for applying maintenance PTFs. These jobs are used when a maintenance tape is applied. For more information on applying maintenance, refer to **Chapter 4, "Applying Maintenance"**.

CSS Jobs

- $CSM0RCV
- $CSM1RCD
- $CSM2APL
- $CSM3ACC

LMS Jobs

- $LMM0RCV
- $LMM1RCD
- $LMM2APL
- $LMM3ACC

Base Services Jobs

- $DVM0RCV
- $DVM1RCD
- $DVM2APL
- $DVM3ACC

HCI Jobs

- $HCM0RCV
- $HCM1RCD
- $HCM2APL
- $HCM3ACC

Step 15. Allocate Compuware Global CSI and SMP/E Datasets

Note: If your site previously installed a Compuware product using SMP/E, skip this step.

This step allocates and initializes the Compuware Global CSI for use with SMP/E. It also allocates the SMPLOG and SMPPTS datasets. For more information on these datasets, refer to Table 2-3 on page 2-5.

The installation facility has created a job to allocate the required datasets based on which components were chosen for installation at your site:

$CSI0GBA (CSS only)

$LMI0GBA (LMS only)

$DVI0GBA (Base Services only)

$HCI0GBA (HCI only)

$ECI0GBA (Two or more components)

1. Review the appropriate job for any necessary changes.
2. Submit the job to start the allocation process.

Step 16. Allocate Libraries and SMP/E Datasets

This step allocates target and distribution zone CSIs as well as product-specific SMP/E datasets.

The installation facility has created a job to allocate the required libraries and datasets based on which components were chosen for installation at your site:

$CSI1PRA (CSS only)

$LMI1PRA (LMS only)

$DVI1PRA (Base Services only)

$HCI1PRA (HCI only)

$ECI1PRA (Two or more components)

Note: The space allocations used are based on 3390 DASD. If you use 3380 DASD, you may decide to change the BLKSIZE for the load library and APF authorized load library to 23476 for a better blocking factor.

1. Review the appropriate job for any necessary changes.
2. Submit the job to start the allocation process.

Step 17. Initialize the Target and Distribution Zone CSIs

This step initializes the CSS, LMS, Base Services, and/or HCI target and distribution zone CSIs.

The installation facility has created a job to initialize the target and distribution zone CSIs based on which components were chosen for installation at your site:

$CSI1PRB (CSS only)

$LMI1PRB (LMS only)

$DVI1PRB (Base Services only)

$HCI1PRB (HCI only)

$ECI1PRB (Two or more components)

1. Review the appropriate job for any necessary changes.

   Note: Before you submit this job, make sure the job submitted in the previous step has completed.

2. Submit the job to start the initialization process.

   Note: This job may return RC=4 if either of the following messages are issued:

   – GIM56501W The aaaaaa subentry was added rather than replaced because it did not exist.

   – GIM27701W aaaaaa entry bbbbbbb was added rather than replaced because it did not exist.

   Otherwise, the return code should be 0. These messages can be ignored.

Step 18. Receive the Base Code and Maintenance PTFs

This step receives the CSS, LMS, Base Services, and/or HCI base code and maintenance PTFs from the SMP/E tape into the SMPPTS and SMPTLIBs.

The installation facility has created a job to receive the base code and maintenance PTFs and to accept the base code depending on which components were chosen for installation at your site:

$CSI2RCV (CSS only)

$LMI2RCV (LMS only)

$DVI2RCV (Base Services only)

$HCI2RCV (HCI only)

$ECI2RCV (Two or more components)

1. Review the appropriate job for any necessary changes.
2. Submit the job to start the initialization process.
3. Review any HOLDDATA listed by the job (if any) and take appropriate actions.
4. This job may return RC=4 in step RCVMAINT if there are no holds to be listed.

CAUTION:

Be sure to run $CSI2RCV, $LMI2RCV, $DVI2RCV, $HCI2RCV or $ECI2RCV in a job class suitable for long-running batch jobs.

Step 19. Apply the Base Code and Maintenance PTFs to the Target Libraries

This step applies the CSS, LMS, Base Services, and/or HCI base code and maintenance PTFs to the target zones.

The installation facility has created a job to apply the base code to the target zone depending on which components were chosen for installation at your site:

$CSI3APL (CSS only)

$LMI3APL (LMS only)

$DVI3APL (Base Services only)

$HCI3APL (HCI only)

$ECI3APL (Two or more components)

1. Review the appropriate job for any necessary changes.
2. Submit the job to Apply the base code and maintenance PTFs.

Note: During maintenance Apply, various libraries may fill up (especially the load libraries), and you will receive the following messages:

```
GIM44001I    SYSTEM ABEND D37 OCCURRED WITH A REASON CODE OF '00000004'X...
GIM43903I    RETRY PROCESSING WILL BE ATTEMPTED...
GIM30302I    COMPRESS PROCESSING WAS SUCCESSFUL...
GIM44001I    SYSTEM ABEND D37 OCCURRED WITH A REASON CODE OF '00000004'X...
GIM43903I    RETRY PROCESSING WILL BE ATTEMPTED...
GIM30302I    COMPRESS PROCESSING WAS SUCCESSFUL...
```

As long as the 3rd message indicates compressing was successful, then these messages can be ignored.

CAUTION:
Be sure to run $CSI3APL, $LMI3APL, $DVI3APL $HCI3APL, or $ECI3APL in a job class suitable for long-running batch jobs.

Note: If the HOLDDATA file from Step 18 is not empty, then the Maintenance Apply Step could receive a return code=4. This is normal and can be ignored.

Step 20. Accept the Base Code and Maintenance PTFs to the Distribution Libraries

This step accepts the CSS, LMS, Base Services, and/or HCI base code and maintenance PTFs into the distribution libraries.

The installation facility has created a job to accept the base code into the distribution libraries depending on which components were chosen for installation at your site:

$CSI4ACC (CSS only)

$LMI4ACC (LMS only)

$DVI4ACC (Base Services only)

$HCI4ACC (HCI only)

$ECI4ACC (Two or more components)

1. Review the appropriate job for any necessary changes.

2. Submit the job to Accept the base code and maintenance PTFs.

CAUTION:
Be sure to run $CSI4ACC, $LMI4ACC, $DVI4ACC, $HCI4ACC, or $ECI4ACC, in a job class suitable for long-running batch jobs.

Note: This job may return RC=4 if any of the following messages are issued:

- GIM24701W SMP/E could not obtain link-edit parameters for load module xxxxxxxx for sysmod xxxxxxxx. Defaults were used.
- GIM23903W Link-edit processing for SYSMOD xxxxxxxx was successful for module mmmmmmmm in LMOD llllllll in the ALCXLOAD library. The return code was 04.
- IEW2454W 9203 SYMBOL xxxxxxxx UNRESOLVED. NO AUTOCALL (NCAL) SPECIFIED.

  • IEW2454W 9203 SYMBOL xxxxxxxx UNRESOLVED. NO AUTOCALL (NCAL) SPECIFIED.

During the ACCEPT processing, individual object modules of a PTF are linked into the ALCXLOAD library. ALCXLOAD is not an executable library. Since CSS specifies the linkedit parm, NCAL (meaning “Do not dynamically include modules called external to the one being processed.”), and the fact that some of these modules contain external calls, the binder issues these warning messages. These calls are usually taken care of during the APPLY link edit processing.

These messages can be ignored.

Step 21. Continue Installation and Customization

Now that the component base code and maintenance have been installed, there may be further customization tasks that need to be addressed. Depending on which components you have installed, continue with the following chapters as necessary:

- **Chapter 5, "CSS Customization"**
- **Chapter 6, "LMS Customization"**
- **Chapter 7, "HCI Customization"**

Chapter 4.
Applying Maintenance

IMPORTANT

If the ECC components exist on your system, you can use the enclosed tape as you would a product maintenance tape to update the components with the latest changes. However, any ECC components which have newer release numbers MUST be installed. If the latest updates for ECC components are already installed, you do not need to reinstall them.

This chapter describes the procedure for applying ECC maintenance.

- If you use a Compuware product that utilizes Compuware Shared Services, you should perform all applicable procedures in this chapter related to CSS. These are typically identified by a reference to CSS following the job name.
- Because all Compuware mainframe products utilize the License Management System, you should also perform all applicable procedures in this chapter related to LMS. These are typically identified by a reference to LMS following the job name.
- Base Services is required for Abend-AID Distributed Viewing Support and for Abend-AID for CICS. You should only apply Base Services maintenance if you use Abend-AID Distributed Viewing Support or Abend-AID for CICS. These jobs are identified by a reference to Base Services following the job name.
- HCI is required for Abend-AID Distributed Viewing Support and for Abend-AID for CICS. You should apply HCI maintenance if you use Abend-AID Distributed Viewing Support or Abend-AID for CICS. HCI is also required if you use STROBE, UNIFACE, or XPEDITER/DevEnterprise. These jobs are identified by a reference to HCI following the job name.

Your ECC installation tape includes all maintenance for the included components available as of the day the tape was created. You may request a separate, cumulative maintenance tape from Compuware Technical Support that contains the latest preventive service. You may also obtain individual PTFs from Compuware to address specific problems you report in priority 1 situations. Maintenance is installed using SMP/E.

If you ran the Installation JCL Customization Facility in the initial installation (see “Step 3. Specify ECC Installation Environment Information” on page 3-5), the following jobs have already been customized:

- $CSM0RCV (CSS), $LMM0RCV (LMS), $DVM0RCV (Base Services), or $HCM0RCV (HCI) performs an SMP/E receive of the maintenance tape.

  ($CSM1RCD, $LMM1RCD, $DVM1RCD, or $HCM1RCD performs an SMP/E receive of maintenance from disk.)
- $CSM2APL (CSS), $LMM2APL (LMS), $DVM2APL (Base Services), or $HCM1APL (HCI) performs an SMP/E apply to target libraries.
- $CSM3ACC performs an SMP/E accept to CSS distribution libraries.
- $LMM3ACC performs an SMP/E accept to LMS distribution libraries.
- $DVM3ACC performs an SMP/E accept to Base Services distribution libraries.
- $HCM3ACC performs an SMP/E accept to HCI distribution libraries.

The following steps are recommended for the maintenance cycle:

1. Accept any PTFs from a previous maintenance run that have not been accepted yet.
2. Receive and Apply the new maintenance PTFs.
3. Test the new PTFs for any problems that may arise due to site dependencies, then move them into production and accept the new PTFs. You could also wait until the next maintenance run before doing the accept.

Note: The ECC maintenance tape includes a separate file that contains updated PTF abstracts ($LCXABS, $LMSABS, $KD1ABS, $HC1ABS). See "Step 1b. Unload PTF Abstracts from Maintenance Tape" on page 4-3 to execute job $RSTABS, which unloads these members into the SMP/E installation sample library.

CAUTION:

Products using address spaces or server regions where CSS, HCI, or Base Services libraries are allocated must be shut down while maintenance is being applied to their CSS libraries and restarted after the maintenance process is complete. This applies specifically to Abend-AID for CICS and XPEDITER/CICS users.

For more information about using SMP/E, consult the *SMP/E Reference* or the *SMP/E User's Guide* for the release of SMP/E you are using.

Step 1a. Unload the ECC SMP/E Installation Sample Library

This step is applicable only if you no longer have the SMP/E installation sample library available. If you still have the library from the initial installation, you may skip to Step 1b.

The SMP/E installation sample library contains JCL samples for completing the installation process. The JCL shown in Figure 4-1 will unload this library. It is unloaded as part of the installation process, see "Step 1. Unload the ECC Installation Sample Library" on page 3-2. This library is contained on the second file of the maintenance tape.

Note: The ttttt should be replaced by the maintenance tape volume serial number and vvvvvv should be replaced by a valid DASD volume serial number. For KCWnnn, nnn should be replaced by the ECC release number.

After you unload the SMP/E installation sample library, execute the Installation JCL Customization Facility to build the following maintenance jobs:

- $CSM0RCV (CSS), $LMM0RCV (LMS), $DVM0RCV (Base Services), $HCM0RCV (HCI)
- $CSM0RCV (CSS), $LMM1RCD (LMS), $DVM1RCD (Base Services), or $HCM1RCD (HCI)
- $CSM2APL (CSS), $LMM2APL (LMS), $DVM2APL (Base Services), or $HCM2APL (HCI)
- $CSM3ACC (CSS), $LMM3ACC (LMS), $DVM3ACC (Base Services), or $HCM3ACC (HCI)

Figure 4-1. JCL To Unload ECC Maintenance Tape

```
//ECC1A    JOB..your job card here...
//UNLOAD   EXEC PGM=IEBCOPY
//SYSUT3   DD UNIT=SYSDA,SPACE=(TRK,5)
//SYSUT4   DD UNIT=SYSDA,SPACE=(TRK,5)
//SYSPRINT DD SYSOUT=*
//TAPE     DD DSN=COMPWARE.LKCW001.F1,
//            UNIT=CART,                          <==VERIFY
//            DISP=(SHR,KEEP),
//            LABEL=(2,SL),
//            VOL=SER=tttttt                      <==VERIFY
//INSTLIB  DD DSN=COMPWARE.KCWnnn.INSTALL,        <==VERIFY
//            UNIT=SYSDA,
//            VOL=SER=vvvvvv,                     <==VERIFY
//            SPACE=(TRK,(45,15,20)),
//            DCB=(BLKSIZE=3120,LRECL=80,RECFM=FB),
//            DISP=(,CATLG)
//SYSIN    DD *
  COPY  INDD=TAPE,OUTDD=INSTLIB
/*
```

Step 1b. Unload PTF Abstracts from Maintenance Tape

If you executed Step 1a, you may skip to Step 2.

This step restores the most current version of the PTF abstracts to the SMP/E installation sample library.

Note: If the $RSTABS member is **NOT** in the sample library, execute Step 1a to unload the sample library to a temporary PDS. Then copy members $RSTABS, $LCXABS, $LMSABS, $KD1ABS, $KMPABS, and $HCMPABS to your SMP/E installation sample library and continue with Step 2.

Update member $RSTABS (as shown in Figure 4-2) with the maintenance tape volser number and, if necessary, the dataset name of the SMP/E Installation Sample library. Submit the job to restore the PTF abstracts into the installation library.

Figure 4-2. $RSTABS JCL

```
//jobname JOB (jobcard information)
//*
//* CHECK THAT THE OUT1 DSN BELOW POINTS TO THE INSTALLATION PDS
//* REPLACE 'TTTTTT' WITH THE MAINTENANCE TAPE VOLSER
//* REPLACE 'NNN' WITH THE ECC RELEASE NUMBER
//*
//RSTRABS     EXEC PGM=IEBCOPY  UNLOAD PTF ABSTRACTS
//SYSPRINT    DD SYSOUT=*
//SYSUDUMP    DD SYOUT=*
//INI         DD DSN=COMPWARE.LKCW001.F5,
//               UNIT=CART,
//               EXPDT=98000,
//               VOL=(,RETAIN,SER=TTTTTT),
//               LABEL=(6,SL),
//               DISP=(,KEEP)
//OUT1       DD  DISP=SHR,DSN=COMPWARE.KCWnnn.INSTALL
//SYSIN      DD  *
  COPY  OUTDD=OUT1,INDD=((IN1,R))
/*
```

Step 2. Receive ECC Maintenance

Use SMP/E Installation Sample Library member $CSM0RCV (CSS), $LMM0RCV (LMS), $DVM0RCV (Base Services), or $HCM0RCV (HCI) to receive any maintenance from the maintenance tape. Review this JCL and change *tttttt* to match the maintenance tape volume serial number, if necessary. $CSM1RCD (CSS), $LMM1RCD (LMS), $DVM1RCD (Base Services), or $HCM1RCD (HCI) will receive maintenance from disk datasets.

After you review and modify the JCL, submit the job to start the receive process.

Note: If the job completes with RC=4, review the output for the following message:

```
GIM24801W NO SYSMODS SATISFIED THE OPERANDS SPECIFIED ON THE
RECEIVE COMMAND
```

If this message appears, no preventive service is applicable for the specified FMIDs.

Step 3. Review HOLDDATA

At times, installing a maintenance tape requires additional actions that must be performed after a PTFs is applied. In these cases, the PTF is identified as an exception with:

```
++HOLD
```

and the HOLDDATA contains information on the additional actions you must perform to completely implement the change.

It is very important that you perform any HOLDDATA action. Failure to do so can cause unpredictable results, including incorrect data and abends. It is possible that the maintenance tape does not have any hold actions. In this case, the HOLDDATA dataset will be empty.

Note: Jobs $CSM0RCV (CSS), $LMM0RCV (LMS), $DVM0RCV (Base Services), or $HCM0RCV (HCI) issue a command to display any HOLDDATA on the maintenance tape.

Step 4. Apply ECC Maintenance

Use the SMP/E installation sample library member $CSM2APL (CSS), $LMM2APL (LMS), $DVM2APL (Base Services), or $HCM2APL (HCI) to apply any maintenance to the target libraries.

Notes:

1. Run the job in a job class suitable for long-running batch jobs.
2. You may be required to have SMP/E restore a USERMOD prior to applying a PTF to a module. SMP/E messages inform you when this action is necessary. After the PTF is applied, you must then reapply the USERMOD.
3. It is advisable to run an APPLY CHECK before applying any maintenance.

The following parameter must be added to the APPLY command of $CSM2APL, $LMM2APL, $DVM2APL, or $HCM2APL to bypass any HOLDDATA: BYPASS(HOLDUSER(ACTION)).

After you review and modify the JCL, submit the job to start the apply process.

Step 5. Accept ECC Maintenance

CAUTION:

Compuware strongly recommends that you do not accept any PTFs until they have been thoroughly tested at your site.

After a function or maintenance (PTFs or APARs) has gone through the ACCEPT step, that becomes your current release-base level. At any time prior to the ACCEPT step, you may issue a RESTORE command to reset your target libraries to the way they were as of the last ACCEPT command. The RESTORE command removes maintenance, or even entire functions, while the ACCEPT command effectively sets a new base level of the release to which further maintenance will be applied. For more information about using the RESTORE command, refer to your site's IBM SMP/E documentation.

Use the SMP/E installation sample library member $CSM3ACC (CSS), $LMM3ACC (LMS), $DVM3ACC (Base Services), or $HCM3ACC (HCI) to have SMP/E accept any maintenance to the distribution libraries. The following parameter must be added to the ACCEPT command of $LMM3ACC or $CSM3ACC to bypass any HOLDDATA: BYPASS(HOLDUSER(ACTION)). Jobs $DVM3ACC or $HCM3ACC may require the parameter BYPASS(HOLDSYSTEM) to bypass any HOLDDATA. Submit the job to start the accept process.

Notes:

1. Run the job in a class suitable for long-running batch jobs.
2. It is advisable to run an ACCEPT CHECK before accepting the **PTFs**.
3. Do **NOT** perform an ACCEPT on any **USERMODs**!

Step 6. Continue ECC Customization

If the Compuware product you are installing also requires customization of Compuware Shared Services (CSS), and/or Distributed Viewing Support (DVS) which requires that Base Services be installed, continue with **Chapter 5, "CSS Customization"**. If the Compuware product you are installing does not utilize CSS, continue with the procedures in **Chapter 6, "LMS Customization"**.

Resolving Problems

If any error messages were returned while performing steps in this chapter, refer to the *Enterprise Common Components Messages and Codes* guide.

If problems persist, contact Compuware Technical Support.

Chapter 5.
CSS Customization

The procedures in this chapter are used to customize Compuware Shared Services (CSS) and the Compuware Viewing Facility (Compuware/VF).

Note: If you are installing Abend-AID with Distributed Viewing Support or Abend-AID for CICS, you need to install Base Services and HCI. For information about configuring the Base Services or HCI server, refer to the *Abend-AID User/Reference Guide* or the *Abend-AID for CICS Installation and Customization Guide.*

For other products needing information about customizing the Host Communications Interface (HCI), refer to **Chapter 7, "HCI Customization"** and **Chapter 8, "HCI Facilities"**.

As shown in Table 5-1, some of the steps in this chapter are required, while others are optional.

Table 5-1. CSS Procedure Requirements

Step Number	Required	Optional
"Step 1. Implement the Security Exit Program (Optional)"		x
"Step 2. Install Customized Translation Tables (Optional)"		x
"Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)"	x	
"Step 4. Activate Compuware ISPF Dialogs (Required)"	x	
"Step 5. Link Compuware/VF and CSS Utilities Tutorials to Your Main Tutorial Panel (Optional)"		x
"Step 6. Associate Contact Information File with Compuware/VF (Optional)"		x
"Step 7. Establish Access to File-AID (Optional)"		x
"Step 8. Prepare the DDIO Files"	x	
"Step 9. Implement the Language Processor JCL (Required)"	x	

Step 1. Implement the Security Exit Program (Optional)

The Security Exit program is an optional user-coded module for establishing security at each site. If you want to ensure full security access to file resources, refer to the "CSS Security Exit" chapter in the Compuware Shared Services User/Reference Guide for detailed information on using the program. A sample Security Exit program is located in SLCXCNTL sample library member CWASSECU.

If the Security Exit program is present, it is called for each member selected. The Security Exit program determines if the user command is permitted, and passes the appropriate return code. If the command is unacceptable for one of the selected members, it will not be executed at all.

Because CSS is used by more than one Compuware product, the Security Exit program is executed for each Compuware product that uses CSS. If you are using multiple Compuware products, you must update your Security Exit program to accommodate those products.

Notes:

1. The sample Security Exit program is for illustration purposes only and **does not** provide any useful functions. You must update it to serve the needs of your site.
2. This Security Exit program is invoked in the MVS environment and does not provide for CICS services.
3. If CICS Abend-AID/FX or Abend-AID for CICS is installed at your site, or if you are installing Abend-AID for CICS, the Security Exit program must be reentrant.

To Implement the Security Exit Program:

Use the following procedure to implement the CWASSECU Security Exit program:

1. Create the Security Exit program. Use the sample Security Exit program as the basis for your exit.

   You must use the program name CWASSECU because other programs reference it and search specifically for this program name.
2. Assemble and linkedit the Security Exit program as a separate load module named CWASSECU. Linkedit CWASSECU into the same load library as CSS. The SLCXCNTL library member CXJCLSEC contains sample JCL that can be used to assemble and linkedit CWASSECU to the CSS load library.
3. If CICS Abend-AID/FX or Abend-AID for CICS is installed at your site, the Security Exit program must be reentrant.

Step 2. Install Customized Translation Tables (Optional)

Note: If you omit this step, the default translation tables will be used.

You must perform this step if you are using Japanese language support with Abend-AID.

You may optionally perform this step if you have Abend-AID with the following product facilities:

- Abend-AID
  - Basic Support
  - with Compuware/VF
  - with Compuware/VF and Compuware language processor

When necessary, Abend-AID products dump storage in both hexadecimal and character representation. These storage areas will appear in one of two formats:

- Horizontal dump
- Vertical dump

The character representation for either format is controlled by a default translation table. The default character set is mixed-case English. You can override either, or both, of these default translation tables with customized translation tables. Different customized translation tables can be specified for either dump format. These customized tables may also allow display of certain otherwise non-displayable fields.

If you install vertical and/or horizontal customized translation tables, CWCMTRVT must be used as the load module name for the vertical translation table, and CWCMTRHT must be used as the load module name for the horizontal translation table. A different internal name can be defined to identify a table. The internal table name must be eight characters long. The table length, excluding the table name, must be 256 bytes. Therefore, the total table length is 264 bytes. If an error is made installing a customized translation table, the Compuware default table is used.

The following SLCXCNTL sample library members contain sample translation tables:

CWCMTRVT	Vertical translation table for uppercase English.
CWCMTRHT	Horizontal translation table for uppercase English.
CWCMTRVE	Vertical translation table for mixed-case English. This is the default vertical translation table that will be used if you choose not to install a customized vertical translation table.
CWCMTRHE	Horizontal translation table for mixed-case English. This is the default horizontal translation table that will be used if you choose not to install a customized horizontal translation table.
CWCMTRVU	Vertical translation table for mixed-case English with the Euro Character X‘9F’ included.
CWCMTRHU	Horizontal translation table for mixed-case English with the Euro character X’9F’ included.

CSS SLCXCNTL sample library member CXJCLTRT contains the sample JCL to assemble and link-edit customized translation tables.

Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)

You must perform this step.

Ensure that the new CSS load modules can be accessed by Compuware products by using one or more of the following methods. Regardless of the method chosen, the CSS load library (SLCXLOAD or a copy of it) should be concatenated **in front** of any Compuware product library(ies). The method you choose will depend on the Compuware products you have at your site.

- Compuware recommends that you place the CSS load library in the link list. This will provide access to the CSS load modules for TSO users and batch jobs that need CSS (for example, language processors, CSS utilities, and/or Abend-AID). This also minimizes any JCL changes that might be needed for a CSS upgrade. (A refresh of the link list is required to activate this change.) Refer to "APF Authorization" on page 2-17 for additional information and cautions about APF authorizations.
- If you have an older version of XPEDITER/CICS installed at your site, you may need to place the CSS load library in the DFHRPL concatenation for each CICS region ahead of the XPEDITER/CICS libraries.
- You may also place the CSS load library in the STEPLIB or ISPLLIB for individual TSO logon PROCs, and in the STEPLIB or JOBLIB for batch jobs. These library specifications will override the CSS load library specified in the link list. This is useful for testing a new version of CSS prior to placing it into the link list for production use.

Notes:

1. If you have previously installed CSS and referenced the load library in batch jobs, compile PROCs, CLISTs, or logon PROCs, CSS will be loaded from the STEPLIB, ISPLLIB, or JOBLIB where it is specified rather than from the link list. Compuware recommends that you review these locations and remove or modify the CSS load library DDs as appropriate. If you have any Compuware product that uses CSS from a STEPLIB, ISPLLIB, or JOBLIB concatenation, you should include the CSS load library in front of the Compuware product in that library concatenation.

   Beginning with CSS 6.0, you have the ability to dynamically allocate the CSS load library as part of the ISPLLIB concatenation. Please refer to Option 1 listed in the next step for more information.

2. For Abend-AID for CICS, specify the CSS, Base Services, and HCI load library names in the server JCL. Refer to the *Abend-AID for CICS Installation and Customization Guide* for more information.

3. If you use the Abend-AID Distributed Viewing Support feature, you must make both the Base Services (SKMPLOAD) and HCI (SLHCLOAD) load libraries available to the Compuware Viewing Facility either in your TSO logon PROC or CLIST, or dynamically when you start the Viewing Facility. See "Option 1 — Dynamically Allocate Libraries when Compuware/VF or CSS Utilities is Executed" on page 5-5 for more information.

4. If you plan to use the Compuware Viewing Facility to view an Abend-AID report, you should use a CSS release that is the same as or more current than the CSS release that was used to write the report or source listing. Using an older CSS release at view-time may cause unpredictable results.

Step 4. Activate Compuware ISPF Dialogs (Required)

You should perform this step if you plan to use the Compuware Viewing Facility and/or CSS Utilities.

Compuware ISPF dialogs include the Compuware Viewing Facility (Compuware/VF) and CSS Utilities. This section covers two different ways to activate the Compuware/VF and CSS Utilities libraries—dynamically, or at TSO startup.

Dynamically allocating the libraries is discussed in Option 1 below. Allocating the libraries at TSO startup is discussed in "Option 2 — Allocate the Libraries at TSO Startup" on page 5-6. For more information about Compuware/VF or CSS Utilities functionality, see the *Compuware Shared Services User/Reference Guide.*

Note: To enable use of the Abend-AID HOTKEY function, you must set up access to either the REXX EXEC or CLIST listed in Table 5-2 on page 5-6 for the Compuware/VF. Follow the instructions detailed in this section for their installation and allocation. Further information on the use of the HOTKEY function is available in the *Abend-AID Installation and Customization Guide* or *Abend-AID User/Reference Guide*. The HOTKEY function can only be invoked while Abend-AID is active, after an abend has occurred, and while it is being displayed on an ISPF screen.

Option 1 — Dynamically Allocate Libraries when Compuware/VF or CSS Utilities is Executed

To dynamically allocate the Compuware/VF and CSS Utilities libraries, you must determine how to make the REXX EXECs and CLISTs accessible.

CSS provides sample REXX EXECs and CLISTs for invoking Compuware/VF and CSS Utilities in SLCXCNTL sample library. You can allocate this library at TSO logon time, or make the CLIST or REXX EXEC accessible in a user-defined library allocated at TSO logon.

IMPORTANT

If CSS REXX EXECs or CLISTs are currently used to invoke Compuware/VF, you must replace them with the latest REXX EXECs or CLISTs in order to use the new Abend-AID features.

Complete the following steps:

1. Determine whether to use REXX EXECs or CLISTs.

   Compuware recommends using REXX EXECs; you may find them easier to customize and maintain.

2. Determine whether Japanese and English language support is required, or just English language support.

3. If using REXX EXECs, choose the appropriate REXX EXEC for Compuware/VF and the appropriate REXX EXEC for CSS Utilities using Table 5-2 as a guide. If using CLISTs, select one CLIST for Compuware/VF and one CLIST for CSS Utilities.

4. Change the library names in the REXX EXECs or CLISTs selected in Step 3 to match your site standards.

   Note: Effective with CSS Release 7.9, an optional ISPSLIB can be used with On-the-Fly processing. See Appendix C in the *Compuware Shared Services User/Reference Guide* for details about On-the-Fly processing. You may use an existing skeleton library allocated at TSO logon time, or allocate one at viewer startup

by updating the REXX EXECs or CLISTs selected in "Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)" on page 5-4

Table 5-2. REXX EXECs or CLISTs Used to Invoke Compuware/VF

Use this REXX EXEC	Or this CLIST	For
CWVFREXE	CWVFCLSE	Compuware/VF, English-only language support, view Abend-AID reports or run utilities
CWVFREXJ	CWVFCLSJ	Compuware/VF, English and Japanese language support, view Abend-AID reports or run utilities
CWUTREXE	CWUTCLSE	CSS Utilities, English-only language support, set up or run XPEDITER utilities
CWUTREXJ	CWUTCLSJ	CSS Utilities, English and Japanese language support, set up or run XPEDITER utilities

Note: When testing a new install of Abend-AID and the new Abend-AID load library has not yet been placed in the link list, it must be listed in CWVFREXE or CWVFCLSE (and/or the Japanese versions) under the ISPLLIB concatenation. Doing this will prevent text merge messages when viewing an Abend-AID report.

5. Allocate the CSS SLCXCNTL library at TSO logon time, or copy the edited REXX EXECs/CLISTs to a library allocated at TSO logon:

   – **To allocate the CSS SLCXCNTL library at TSO logon**

   Add the library to the SYSPROC concatenation in the TSO logon PROC.

   – **To make specific REXX EXECs/CLISTs accessible in a library allocated at TSO logon**

   Copy the REXX EXECs or CLISTs to an existing SYSPROC library allocated at TSO logon.

6. Users can invoke either Compuware/VF or CSS Utilities by selecting an option from an ISPF panel (recommended), or by executing a REXX EXEC or CLIST.

   – **To enable users to select an ISPF menu option**

   Add one or more of the following lines to an ISPF panel. Note that 'o' represents the option to be entered at the menu command prompt.

To invoke Compuware/VF	o,'CMD (%*Compuware/VF CLIST name* ***or*** *Compuware/VF REXX EXEC name*)
To invoke CSS Utilities	o,'CMD (%*CSS Utilities CLIST name* ***or*** *CSS Utilities REXX EXEC name*)

   – **To invoke Compuware/VF or CSS Utilities by executing a REXX EXEC or CLIST**

   Enter the appropriate command at the TSO command prompt:

To invoke Compuware/VF	TSO %*Compuware/VF CLIST name* ***or*** *Compuware/VF REXX EXEC name*
To invoke CSS Utilities	TSO %*CSS Utilities CLIST name* ***or*** *CSS Utilities REXX EXEC name*

Option 2 — Allocate the Libraries at TSO Startup

Allocating the load, panel, and message libraries at TSO logon simplifies maintenance. All TSO users have access to Compuware/VF and CSS Utilities after logging on to TSO, and upgrades that affect all TSO users can be made in one central location. Allocating

these libraries at TSO logon may be inappropriate, however. In that case, dynamically allocating the libraries may be a better solution.

To provide access to the load, panel, and message modules at TSO logon time, complete the following steps.

1. Concatenate each library as shown in Table 5-3:

Table 5-3. Allocating CSS Libraries

Load library SLCXLOAD	Place the CSS load library in the link list (see "Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)" on page 5-4 for instructions) OR Place a DD statement for the load library in the concatenation for STEPLIB or ISPLLIB in your TSO logon PROC OR Execute a CLIST or REXX EXEC at logon time that will concatenate the load library to the appropriate DD statement before starting ISPF
Load library SKMPLOAD Base Services — required for Abend-AID Distributed Viewing Support or Abend-AID for CICS	Place the Base Services load library in the link list (see "Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)" on page 5-4 for instructions) OR Place a DD statement for the load library in the concatenation for STEPLIB or ISPLLIB in your TSO logon PROC OR Execute a CLIST or REXX EXEC at logon time that will concatenate the load library to the appropriate DD statement before starting ISPF **Note:** Abend-AID for CICS users must place a DD statement for the load library in the concatenation for FDBDRPL in your server startup JCL.
Load library SLHCLOAD HCI — required for Abend-AID Distributed Viewing Support or Abend-AID for CICS	Place the HCI load library in the link list (see "Step 3. Make New CSS Load Modules Accessible to Compuware Products (Required)" on page 5-4 for instructions) OR Place a DD statement for the load library in the concatenation for STEPLIB or ISPLLIB in your TSO logon PROC OR Execute a CLIST or REXX EXEC at logon time that will concatenate the load library to the appropriate DD statement before starting ISPF **Note:** Abend-AID for CICS users must place a DD statement for the load library in the concatenation for FDBDRPL in your server startup JCL.
Message library SLCXMENU	Concatenate the library to ISPMLIB in the TSO logon PROC OR Concatenate the library to ISPMLIB in a CLIST or REXX EXEC before starting ISPF
Panel library SLCXPENU	Concatenate the library to ISPPLIB in the TSO logon PROC OR Concatenate the library to ISPPLIB in a CLIST or REXX EXEC before starting ISPF
EXEC library SLCXEXEC	Concatenate the library to SYSPROC in the TSO logon PROC OR Concatenate the library to SYSPROC in a CLIST or REXX EXEC before starting ISPF

Table 5-3. Allocating CSS Libraries

Skeleton Library ISPSLIB	Optional library for On-the-Fly processing. See Appendix C in the *Compuware Shared Services User/Reference Guide* for details. Concatenate the library to ISPSLIB in the TSO logon PROC OR Concatenate the library to ISPSLIB in a CLIST or REXX EXEC before starting ISPF.

Notes:

a. Replace any existing CSS libraries in these concatenations.

b. The low level qualifiers are listed below their respective library names.

c. When testing a new install of Abend-AID and the new Abend-AID load library has not yet been placed in the link list, it must be added as explained in the instructions for the SLCXLOAD library. Doing this will prevent text merge messages when viewing an Abend-AID report.

2. Add one or more of the following lines to an ISPF panel. Note that 'o' represents the option to be entered at the ISPF menu command prompt.

To invoke Compuware/VF	o,'PGM(CWDDSUTL) NEWAPPL(AAUT) PARM(SPF)'
To invoke CSS Utilities	o,'PGM(CWDDIUFE) NEWAPPL(AAUT)'

Step 5. Link Compuware/VF and CSS Utilities Tutorials to Your Main Tutorial Panel (Optional)

CSS SLCXCNTL sample library contains two sample tutorial main menu panels for ISPF and ISPF/PDF:

- CXTUTOR
- CXR00003

To link the Compuware/VF and CSS Utilities tutorial panels to your tutorial main menu panel (TTUTOR or ISR00003), use the applicable member as an example.

Step 6. Associate Contact Information File with Compuware/VF (Optional)

The Contact Information feature enables Compuware/VF users to associate contact information (for example, the phone number and e-mail address of an off-duty programmer) with a particular job name. In order to use this feature, the contact information file must be associated with Compuware/VF.

Complete the following steps:

1. Use the following members from the CSS SLCXCNTL sample library:
   - CXCIVSAM
   - CXCIREG
2. Edit member CXCIVSAM and modify the COMPWARE.CONTACT.INFO dataset name to conform to your installation's naming standards.
3. Execute CXCIVSAM to allocate the VSAM dataset that will store contact information.
4. Edit member CXCIREG and change the CDSN symbolic parameter to the dataset name used in Step 2.
5. Set the CXLOAD symbolic parameter to the CSS load library dataset name.
6. Set the CXCNTL symbolic parameter to the CSS SLCXCNTL sample JCL dataset name.
7. (Optional) Specify the UNIT and ASSEMBLER symbolic parameters.
8. Execute CXCIREG to register the contact information dataset.
9. To populate the Contact Information file, refer to the *Compuware Shared Services User/Reference Guide* "Working with Contact Information" section in the Compuware Viewing Facility chapter.

If you wish to use a Contact Information dataset created prior to CSS 7.8 (PTF level CXF0168), it must be converted into the new extended format. The CSS SLCXCNTL sample library contains the CXCIVSAM and CXCICONV members.

1. Edit the CXCIVSAM member and change the dataset name to a new name. Execute the job.
2. Edit the CXCICONV member. Change the CIDSN symbolic parameter to point to the original contact information dataset.
3. Change the CODSN symbolic parameter to point to the new dataset you just allocated.
4. Execute CXCICONV. Follow the instructions for step 9 above and adjust the Contact Information data if necessary.

Step 7. Establish Access to File-AID (Optional)

File-AID users may now access File-AID/MVS, File-AID for DB2, or File-AID for IMS directly from Compuware/VF using simple line commands. Installation requirements for each version of File-AID are listed below. Abend-AID release 9.1 or more current must also be installed.

File-AID/MVS Users

To access File-AID/MVS from the Compuware/VF Output Selection Menu Option 8, complete the following steps:

1. Ensure that File-AID/MVS 8.5 or more current is installed.
2. Either:
   - Allocate the File-AID/MVS libraries at TSO logon

     OR
   - Locate the CWVFCLSE CLIST or CWVFREXE REXX EXEC in the SLCXCNTL sample library. Uncomment the lines labeled **FA/MVS PANEL LIB**, **FA/MVS MESSAGE**, **FA/MVS LOADLIB**, and **FA/MVS LOADLIB** and replace the existing library names with your actual File-AID/MVS library names.

File-AID for DB2 Users

To access File-AID for DB2 from the Compuware/VF Output Selection Menu Option 11, complete the following steps:

1. Ensure that File-AID for DB2 V3R9 or more current is installed.
2. Locate the CWVFDB2 and FDXTRN01 CLISTs in the SLCXCNTL sample library.
3. Move these CLISTs to a library allocated at TSO logon.

   If Compuware/VF is started by executing CWVFCLSE or CWVFREXE you can place these CLISTs into the SLCXEXEC library.
4. Locate the following line in CLIST CWVFDB2:

   ```
   /* SET &FADB2CL = '????????.FADB2.ISRCLIB'
   ```

   Replace this dataset name with the name of the File-AID for DB2 CLIST library.
5. (Optional) To dynamically allocate the File-AID for DB2 ISPF and CLIST libraries when a user accesses File-AID from Compuware/VF, set the LIBDEF PROC variable in CLIST CWVFDB2 to **Y**.

File-AID for IMS Users

To access File-AID for IMS from the Compuware/VF Output Selection Menu Option 9, the following conditions must be met:

- File-AID for IMS 4.7 with product update F1 or more current must be installed.
- File-AID for IMS CLISTs FIALLOC and FIFREE must exist in a CLIST allocated to the user's SYSPROC DD.

Note: If any File-AID error messages are received within the Compuware/VF while viewing any of the sections listed above, and access to the required File-AID libraries has been set up properly, please contact your File-AID support representative.

Step 8. Prepare the DDIO Files

You must perform this step if you are a new CSS user or have never created DDIO files and now need to use them. If you are an Abend-AID user and are installing or migrating to Release 9.4 or later you will need to allocate a Report Shared Directory and one or more Report databases.

CSS DDIO files are used to store diagnostic reports, transaction reports, and source listings. The term *DDIO file* is a generic term used to refer to the datasets that store the reports and listings from Compuware Products that use CSS. This step describes the various types of DDIO files available and who needs to allocate which type.

Types of DDIO Files

There are two main categories of DDIO files: report or source listing files and report or source listing *databases* which are grouped into pools that are attached to Shared Directories. CSS uses the Shared Directory architecture to manage the available pool space.

Within these two categories of DDIO files, there are three different types of DDIO files characterized by the type of information that is stored in them: abend reports, source listings, and CICS transaction reports.

Note: If you use the Shared Directory architecture for your DDIO files, different database types cannot be attached to the same Shared Directory (for example, you cannot attach a source listing database to a report shared directory, you cannot attach a transaction database and report database to the same Shared Directory, etc.).

Note: Non-database report or source listing files cannot be attached to a Shared Directory until they have been converted. See the "Batch File Utility CWAASDUT" and the "Batch File Utility CWDDLPUT" chapters in the *CSS User/Reference Guide* for more information.

1. Abend Reports:
   a. **Report Shared Directories and Databases.** This type of file is used to store abend reports created by Abend-AID. *This Report Shared Directory architecture is only supported for Abend-AID Release 9.4 and later.* Report databases are attached to a report shared directory. The report databases are managed as a POOL of space belonging to the report shared directory to which they are attached.
   b. **Report DDIO files.** This type of DDIO file is used to store Abend Reports created by Abend-AID. Prior to Abend-AID Release 9.4, this was the only type of report file available. For compatibility purposes, Abend-AID Release 9.4 can also use this type of report file. A conversion utility is available to run when the user is ready to migrate the report file to the report database format.
2. Source Listings:
   a. **Source Listing Shared Directories and Databases.** This type of DDIO file is used to store compiler source listings for use by various Compuware Products. Source listing databases are attached to a source listing shared directory. The source listing databases are managed as a POOL of space belonging to the source listing shared directory to which they are attached. *The Source Listing Shared Directory architecture can now be used with all supported language compilers. However, XPEDITER users should see notes and exceptions below.*
   b. **Source Listing DDIO Files.** This type of DDIO file is used to store compiler source listings. A conversion utility is available to run when the user is ready to migrate the listing file to the source listing database format.

The same source listing file, database, or source listing member can be used by multiple Compuware Products. It is not necessary to create duplicate source listing files, databases, or source listing members for each Compuware Product.

3. **Transaction Databases and Shared Directory for Abend-AID for CICS.** Transaction databases are used by Abend-AID for CICS to store CICS transaction dumps. Abend-AID for CICS catalogs both CICS transaction dumps and imported Region Dumps into the shared directory.

The following table lists the types of DDIO files created by the CSS utilities and which products use/require them:

Table 5-4. DDIO File Formats

Compuware Product	Report File	Source Listing File	Transaction Databases & Shared Dir.	Report Shared Dir. & Database	Source Listing Shared Dir.	Source Listing Database
Abend-AID	x	x		x	x	x
Abend-AID for CICS		x	x		x	x
XPEDITER/TSO		x			x[1] See Note	x
XPEDITER/IMS		x			x[1] See Note	x
XPEDITER/CICS		x			See Note	x

[1] If using XPEDITER/TSO & XPEDITER/IMS with the C Language Processor, VisualAge PL/I, or Enterprise PL/I, and you use Long Program Names, the Source Listing Shared Directory must be used.

Note: XPEDITER/TSO Release 7.3, XPEDITER/CICS Release 7.6 and Abend-AID for CICS Release 4.5 support source listing shared directories. Prior releases of these products **do not** support source listing shared directories. However, individual source listing databases **can** be used by all CSS-enabled Compuware products if the databases are explicitly referenced. See footnote 1 above for exceptions.

Refer to the "Source Listing Databases and Shared Directories" section of Chapter 2 in the *Compuware Shared Services User/Reference Guide* for more information.

File Access Methods

You may use either sequential or VSAM files for DDIO files. Shared directories must be VSAM files. If you are using VSAM files for your DDIO files, you can copy them using the IDCAMS REPRO command. You cannot, however, change any of the file attributes such as CISIZE without reallocating a DDIO/DB file. Allocation and format requirements, along with the steps for preparing the various DDIO files, will vary according to the type of file. Source listing files can be shared between all Compuware products that use CSS, while report files, transaction databases, work files, and profile files, cannot.

CAUTION:

CWDDSUTL, CWFXSDUT, CWAASDUT, and CWDDLPUT are the only utilities recommended for use with Compuware DDIO files.

File-AID/MVS and other VSAM-related vendor products (including vendor packages that compress or enhance buffer management on VSAM files) should NOT be used against Compuware DDIO files. Doing so can result in corrupting the contents of the DDIO file. Refer to "Chapter 2. Allocating and Formatting DDIO Files" in the *Compuware Shared Services User/Reference Guide* for more information.

Step 8a. Allocate and Format Abend-AID Report Shared Directories and Databases for Abend-AID Release 9.4 and Later Users

This step should only be performed by Abend-AID users who have installed, are installing, or are migrating to Abend-AID Release 9.4 or later. The purpose of this step is to allocate and format a Report Shared Directory and one or more Report Databases attached to that Shared Directory. Note that Abend-AID 9.4 is the minimum release able to use these types of DDIO file.

This step can be performed now, or during the Abend-AID installation.

Create the Report Shared Directory

Use the sample JCL in the CSS SLCXCNTL dataset to allocate and format the Report Shared Directory (member is CXALLDAA). For additional options refer to the "Batch File Utility CWAASDUT" chapter in the *Compuware Shared Services User/Reference Guide*. The shared directory CREATE command will both ALLOCATE and FORMAT the shared directory. If the file currently exists, you must specify parameter REALLOCATE=YES or the allocate will fail with a duplicate file error.

Create the Report Databases

You must have already created the Report Shared Directory before attempting to create the Report Database. You **can** create both in the same job as long as the Report Shared Directory create is done first. The database CREATE command will both ALLOCATE and FORMAT the database. If the file currently exists, you must specify parameter REALLOCATE=YES or the allocate will fail with a duplicate file error.

VSAM Report Databases can be created using the sample JCL in the CSS SLCXCNTL dataset (member is CXALLVAA).

Sequential Report Databases can be created using the sample JCL in the CSS SLCXCNTL dataset (member is CXALLBAA).

Step 8b. Allocate and Format Abend-AID Report Files for Pre-Abend-AID Release 9.4 Users

You **must** perform this step if you are an Abend-AID user and did **not** perform "Step 8a. Allocate and Format Abend-AID Report Shared Directories and Databases for Abend-AID Release 9.4 and Later Users", **and** *do not* have an existing Report File.

Allocate the Report Files

VSAM Report files can be allocated using the sample JCL in the CSS SLCXCNTL dataset (member is CXALLVS, parm member is CXALDDAA).

Sequential Report files can be allocated using the sample JCL in the CSS SLCXCNTL dataset member CXALLDS.

Format the Report Files

After the report files have been allocated, they must be formatted before they can be used. The formatting process initializes the directory structure and specifies the file options.

CSS SLCXCNTL library member CXFMTDS contains sample JCL to format a report file with appropriate defaults for the sample files allocated using members CXALLVS or CXALLDS. Change CXPARM to specify CXAARPT as the member containing the FORMAT control statement for the report file being formatted. You can optionally specify your own customized member in place of CXAARPT.

Step 8c. Allocate and Format Source Listing Shared Directories and Databases

This step may be performed by all users who use Source Listing support. Please note the exceptions for XPEDITER and Abend-AID for CICS users above.

Create the Source Listing Shared Directory.

Use the sample JCL in the CSS SLCXCNTL dataset to allocate and format the Source Listing Shared Directory (member is CXALLDLP). For additional options refer to the "Batch File Utility CWDDLPUT" chapter in the *Compuware Shared Services User/Reference Guide*. The shared directory CREATE command will both ALLOCATE and FORMAT the shared directory. If the file currently exists, you must specify parameter REALLOCATE=YES or the allocate will fail with a duplicate file error.

Create the Source Listing Databases

You must have already created the Source Listing Shared Directory before attempting to create the Source Listing Database. You **can** create both in the same job as long as the Source Listing Shared Directory create is done first. The database CREATE command will both ALLOCATE and FORMAT the database. If the file currently exists, you must specify parameter REALLOCATE=YES or the allocate will fail with a duplicate file error.

VSAM Source Listing Databases can be created using the sample JCL in the CSS SLCXCNTL dataset (member is CXALLVLP).

Sequential Source Listing Databases can be created using the sample JCL in the CSS SLCXCNTL dataset (member is CXALLBLP).

Step 8d. Allocate and Format Source Listing Files

You must perform this step if you did **not** perform "Step 8c. Allocate and Format Source Listing Shared Directories and Databases" **and** you do not have an existing source listing file or database.

Allocate the Source Listing Files

VSAM Source Listing files can be allocated using the sample JCL in the CSS SLCXCNTL dataset (member is CXALLVS, parm member is CXALDDSL).

Sequential Source Listing files can be allocated using the sample JCL in the CSS SLCXCNTL dataset member CXALLDS.

Format the Source Listing Files

After the source listing files have been allocated, they must be formatted before they can be used. The formatting process initializes the directory structure and specifies the file options.

CSS SLCXCNTL library member CXFMTDS contains sample JCL to format a source listing file with appropriate defaults for the sample files allocated using members CXALLVS or CXALLDS. Change CXPARM to specify CXFMTLST as the member containing the FORMAT control statement for the source listing file being formatted. You can optionally specify your own customized member in place of CXFMTLST.

Step 8e. Allocate and Format Abend-AID for CICS Shared Directories and Transaction Databases

Normally this would be performed during the Abend-AID for CICS install procedure. For additional information, see the *Abend-AID for CICS Installation and Customization Guide.*

Create the Abend-AID for CICS Shared Directory (Allocate & Format)

Use the sample JCL to create the shared directory. CSS SLCXCNTL sample library member CXALLMC contains the sample JCL for creating a Abend-AID for CICS shared directory.

Create the Abend-AID for CICS Transaction Database (Allocate & Format)

You must create the shared directory before proceeding to this step. Use one of the following two sets of sample JCL to create each Abend-AID for CICS database as needed:

VSAM Abend-AID for CICS Database: CSS SLCXCNTL sample library member CXALLVSD contains sample JCL for creating a Abend-AID for CICS VSAM transaction database.

Sequential Abend-AID for CICS Database: CSS SLCXCNTL sample library member CXALLBSD contains sample JCL for creating a Abend-AID for CICS sequential transaction database.

Step 9. Implement the Language Processor JCL (Required)

See the sub-steps below to determine when you must perform this step.

The language processors support two methods of processing your programs: preprocessing and postprocessing. Both of these choices may involve modifications to your compile JCL.

You may use the preprocessor or postprocessor (whichever is best for your site as needed.)

This section discusses the benefits of the preprocessor and the postprocessor and describes when to use them. See the Compuware Shared Services User/Reference Guide for more information on determining when to use the pre- or postprocessor.

Refer to “CSS Language Processor (LP)” on page 2-9 for descriptions of preprocessing and postprocessing and a summary of the benefits provided by each method.

To use the language processor, you must have previously completed installation of the CSS load library and also prepared a DDIO file. For each language you are using, you should modify the corresponding compiler procedure to provide for the type of processing you will be implementing.

When running the language processor as a preprocessor, you can specify assembler/compiler options in the CWPPRMO dataset or in the PARM statement passed to the preprocessor. The CWPPRMO is an input dataset that contains the language processor compiler options. Depending on the method used, it may be necessary to include the LANGPARM command with these options.

If you are currently using the postprocessor, and you will be converting to the preprocessor, you are not required to spool your SYSPRINT to a temporary dataset.

Note: For each of the following steps, it is possible that you will receive a return code other than zero. Should this occur, note the accompanying language processor error message and refer to the *Enterprise Common Components Messages and Codes* guide for further information.

Step 9a. Modify the JCL to Run the COBOL Language Processor

You must perform this step if either of the following is true:

- You are installing XPEDITER/CICS for the first time.
- You are installing Abend-AID for the first time or you are upgrading a current Abend-AID installation with Abend-AID source support **and** you are licensed for the Compuware COBOL language processor.

You may optionally perform this step if you are installing Abend-AID for CICS or XPEDITER/TSO **and** you are licensed for the Compuware COBOL language processor. XPEDITER/TSO customers can either perform this step now in batch mode, or online as part of the XPEDITER/TSO installation procedure.

This step contains instructions for both the COBOL language preprocessor and postprocessor. You should follow the instructions that apply to the type of processing you will be performing.

Preprocessor

SLCXCNTL sample library member CXCOBPRE contains sample JCL necessary to run the COBOL language preprocessor.

Modify the compile step of your COBOL JCL:

- On the EXEC statement, change the name of your compiler to CWPCMAIN and add the following to any existing COBOL compiler options:

```
LANGUAGE(compiler version)
```

Notes:

a. The LANGUAGE command may be specified in the CWPPRMO command stream rather than on the EXEC statement.

b. The preprocessor accepts VSCOBOLII, VSCOBOLIIREL3, VSCOBOLIIREL4, COBOL/370, COBOL/390, COBOL/MVS, and COBOLZ/OS as being equivalent because the program name is the same. The compiler that is actually executed depends on the library specified in the STEPLIB, JOBLIB, and/or LINKLIST concatenation.

- Add the CSS SLCXLOAD DSN to the STEPLIB or JOBLIB concatenation if it is not in the link list.
- Add the CWPDDIO DD statement to specify the name of the source listing file (DDIO file) you will be using.
- Determine the language processor options you will use as input and add them to the EXEC parameter, or add a CWPPRMO DD statement that points to the appropriate options. Sample options are contained in SLCXCNTL sample library members CXLPCOBB (for batch programs) and CXLPCOBC (for CICS programs) to provide the smallest output to the source listing file (for minimizing space) and generate a printout of the COBOL enhanced listing. When using these options, a printout of the listing from the DDIO file contains only the source statements. The XREF and other portions of the listing are not written to the DDIO file. If you use the sample options, note the following:
  - If you do not have XPEDITER/TSO, remove DDIO(OUTPUT(FIND)).
  - If the program will be used by XPEDITER/CICS or Abend-AID for CICS, use the options in sample CXLPCOBC.
- The SYSPRINT file will be LRECL=133, RECFM=FBA.

Postprocessor

You can use the postprocessor to process compile listings that were previously stored, or you can modify your COBOL compile JCL to pass the compiler listing to the postprocessor as described below.

- Check the COBOL compiler options.
- Change the COBOL compiler JCL to write the compiler listing (SYSPRINT) to a sequential file that can be used as input to the COBOL language postprocessor. A sample of the necessary JCL is shown below.

```
//SYSPRINT DD UNIT=SYSDA,SPACE=(TRK,(25,20)),DCB=BLKSIZE=16093,
//            DISP=(MOD,PASS)         <===== CA-OPTIMIZER NEEDS 'MOD'
```

  The above example is a temporary file. You can save the JCL as a permanent file to be reused as shown in the example below:

  //SYSPRINT DD DSN=CX,STORED.LISTINGS(MEMBER)

- The SYSPRINT file will be LRECL=133, RECFM=FBA.
- Add SLCXCNTL sample library member CXCOB1 as the postprocessor step following your COBOL compile step.
- Determine the language processor options you will use as input and add them to the EXEC parameter, or add a CWPPRMO DD statement that points to the appropriate options. Sample options are contained in SLCXCNTL sample library members

CXLPCOBB (for batch programs) and CXLPCOBC (for CICS programs) to provide the smallest output to the source listing file (for minimizing space) and generate a printout of the COBOL listing. When using these options, a printout of the listing from the DDIO file will contain only the source statements. The XREF and other portions of the listing will not be written to the DDIO file. If you use the sample options, note the following:

- If you do not have XPEDITER/TSO, remove DDIO(OUTPUT(FIND)).
- If the program will be used by XPEDITER/CICS or Abend-AID for CICS, use the options in sample CXLPCOBC.

The SLCXCNTL sample library members that are supplied for the COBOL language postprocessor invoke the language processor in different ways:

Table 5-5. SLCXCNTL Sample Library Members for COBOL Language Processor

Member	Purpose
CXCOB1	Processes compiler listings through the language postprocessor. Add this JCL after your COBOL compile step and before your link edit step (if one exists).
CXCOB2	Postprocesses compiler listings stored from a previous compile job. Use this JCL to process listings that have been previously compiled and stored in machine-readable format.
CXCOB99	Compiles COBOL programs and feeds the compiler output into the COBOL postprocessor.

- The CWPPRTO file will be LRECL=133, RECFM=FBA.
- The CWPLOAD file must be allocated using DISP=OLD or DISP=SHR.
- The object module that is processed by the postprocessor must be used as input to the linkedit step.

Step 9b. Modify the JCL to Run the PL/I Language Processor

You must perform this step if the following is true:

- You are installing Abend-AID with Compuware/VF, **or** XPEDITER/CICS, **or** XPEDITER/TSO **and** you are licensed for the Compuware PL/I language processor.

This step contains instructions for both the PL/I language preprocessor and postprocessor. You should follow the instructions that apply to the type of processing you will be performing.

Preprocessor

SLCXCNTL sample library member CXPLIPRE contains sample JCL necessary to run the PL/I language preprocessor.

Modify the compile step of your PL/I JCL:

- On the EXEC statement, change the name of your compiler to CWPPMAIN and add the following to any existing PL/I compiler options:

```
LANGUAGE(compiler version)
```

  Refer to the *Compuware Shared Services User/Reference Guide* for a list of the compiler versions that can be specified with the LANGUAGE command. The default compiler for PL/I is the compiler currently in use on your system.

  Note: The LANGUAGE command may be specified in the CWPPRMO command stream rather than on the EXEC statement.

- Add the CSS SLCXLOAD DSN to the STEPLIB or JOBLIB concatenation if it is not in the link list.

- Add the CWPDDIO DD statement to specify the name of the source listing file (DDIO file) you will be using.
- Determine the language processor options you will use as input and add them to the EXEC parameter, or add a CWPPRMO DD statement that points to the appropriate options. Sample options are contained in SLCXCNTL sample library member CXLPPLI to provide the smallest output to the source listing file (for minimizing space). If you use these options, and you will be using the PL/I language preprocessor, add LANGUAGE(xxx) to the EXEC PARM or to the CWPPRMO DD statement.

See the *Compuware Shared Services User/Reference Guide* for more information on this JCL and PL/I language processor options.

Postprocessor

You can use the postprocessor to process compile listings that were previously stored, or you can modify your PL/I compile JCL to pass the compiler listing to the postprocessor as described below.

- Check the PL/I compiler options.
- Change the PL/I compile JCL to write the compiler listing (SYSPRINT) to a sequential file for use as input to the PL/I language postprocessor. A sample of the necessary JCL is shown below.

```
//SYSPRINT DD UNIT=SYSDA,SPACE=(CYL,(100,10)),DCB=BLKSIZE=19069,
//            DISP=(MOD,PASS)
```

  The above example is a temporary file. You can save the JCL as a permanent file to be reused as shown in the example below:

  //SYSPRINT DD DSN=CX,STORED.LISTINGS(MEMBER)

- The SYSPRINT file will be LRECL=133, RECFM=FBA.

Table 5-6. SLCXCNTL Sample Library Members for PL/I Language Processor

Member	Purpose
CXPLI	Processes compiler listings through the language postprocessor. Add this JCL after your PL/I compile step and before your link edit step (if one exists). CXPLI can be customized to meet your site's requirements.
CXPLI2	Postprocesses compiler listings stored from a previous compile job. Use this JCL to process listings that have been previously compiled and stored in machine-readable format.

- The CWPLOAD file must be allocated using DISP=OLD or DISP=SHR.
- The object module that is processed by the postprocessor must be used as input to the linkedit step.

See the *Compuware Shared Services User/Reference Guide* for more information on this JCL and PL/I language processor options.

Step 9c. Modify the JCL to Run the Assembler Language Processor

You must perform this step if you are licensed for the Assembler language processor option for any product.

You can optionally perform this step if you are installing XPEDITER/TSO **and** you are licensed for the Compuware Assembler language processor. XPEDITER/TSO customers can either perform this step now in batch mode, or online as part of the XPEDITER/TSO installation procedure.

This step contains instructions for both the Assembler language preprocessor and postprocessor. You should follow the instructions that apply to the type of processing you will be performing.

SLCXCNTL sample library member CXASMPRE contains sample JCL necessary to run the Assembler language preprocessor.

Preprocessor

Modify the assembly step of your Assembler JCL:

- On the EXEC statement, change the name of your compiler to CWPAMAIN and add the following to any existing assembler options:

  ```
  LANGUAGE(assembler version)
  ```

  Refer to the *Compuware Shared Services User/Reference Guide* for a list of the assembler versions that can be specified with the LANGUAGE command. The default assembler is Assembler H.

  Note: The LANGUAGE command may be specified in the CWPPRMO command stream rather than on the EXEC statement.

- Add the CSS SLCXLOAD DSN to the STEPLIB or JOBLIB concatenation if it is not in the link list.
- Add the CWPDDIO DD statement to specify the name of the source listing file (DDIO file) you will be using.
- Determine the language processor options you will use as input and add them to the EXEC parameter, or add a CWPPRMO DD statement that points to the appropriate options. Sample options are contained in SLCXCNTL sample library member CXLPASM to provide the smallest output to the source listing file (for minimizing space). When using these parameters, a printout of the listing from the DDIO file will contain only the source statements. The XREF will not be written to the DDIO file. If you use these options and you will use the Assembler language preprocessor, add LANGUAGE(xxx) to the EXEC PARM or to the CWPPRMO DD statement.
- Specify the compiler (and release) you will use by including it as a parameter of the LANGUAGE command in your EXECUTE parameter or CWPPRMO DD statement.

Refer to the *Compuware Shared Services User/Reference Guide* for more information on the LANGUAGE command, for a list of available compilers, and for more information on this JCL and language processor options.

Postprocessor

You can use the postprocessor to process compile listings that were previously stored, or you can modify your Assembler compile JCL to pass the compiler listing to the postprocessor as described below.

- Check the assembler options.
- Change your assembler JCL to write the compiler listing (SYSPRINT) to a sequential file for use as input to the Assembler language postprocessor. A sample of the necessary JCL is shown below.

  ```
  //SYSPRINT DD UNIT=SYSDA,SPACE=(TRK,(25,20)),DCB=BLKSIZE=16093,
  //            DISP=(MOD,PASS)
  ```

  The above example is a temporary file. You can save the JCL as a permanent file to be reused as shown in the example below:

```
//SYSPRINT DD DSN=CX,STORED.LISTINGS(MEMBER)
```

- The SYSPRINT file will be LRECL=133, RECFM=FBA.
- Add SLCXCNTL sample library member CXASM as the postprocessor step following your assembler step. CXASM can be customized to meet your site's requirements.

  Note: The CWPLOAD file must be allocated using DISP=OLD or DISP=SHR.

- The object module that is processed by the postprocessor must be used as input to the linkedit step.

Refer to the *Compuware Shared Services User/Reference Guide* for more information on this JCL and language processor options.

Step 9d. Modify the JCL to Run the C Language Processor

You must perform this step if you are installing XPEDITER/TSO and you are licensed for the Compuware C language processor.

This step contains instructions for both the C language preprocessor and postprocessor. You should follow the instructions that apply to the type of processing you will be performing.

Preprocessor

SLCXCNTL sample library member CXCPRE contains sample JCL necessary to run the C language preprocessor.

Modify the compile step of your C JCL:

- On the EXEC statement, change the name of your compiler to CWPZMAIN and add the following to any existing C compiler options:

  ```
  LANGUAGE(compiler version)
  ```

  Refer to the Compuware Shared Services User/Reference Guide for a list of the compiler versions that can be specified with the LANGUAGE command. The default compiler for C is the compiler currently in use on your system.

  Note: The LANGUAGE command may be specified in the CWPPRMO command stream rather than on the EXEC statement.

- Add the CSS SLCXLOAD DSN to the STEPLIB or JOBLIB concatenation if it is not in the link list.
- Add the **CWPDDIO DD** statement to specify the name of the source listing file (DDIO file) you will be using.
- Determine the language processor options you will use as input and add them to the EXEC parameter, or add a **CWPPRMO DD** statement that points to the appropriate options. Sample options are contained in SLCXCNTL sample library member CXLPC to provide the smallest output to the source listing file (for minimizing space). If you use these options, and you will be using the C language preprocessor, add LANGUAGE(xxx) to the EXEC PARM or to the **CWPPRMO DD** statement.

Refer to the *Compuware Shared Services User/Reference Guide* for more information on this JCL and C language processor options.

Postprocessor

You can use the postprocessor to process compile listings that were previously stored, or you can modify your C compile JCL to pass the compiler listing to the postprocessor as described below.

- Check the C compiler options.
- Change the C compile JCL to write the compiler listing (SYSCPRT) to a sequential file for use as input to the C language postprocessor. A sample of the necessary JCL is shown below.

```
//SYSCPRT DD UNIT=SYSDA,SPACE=(CYL,(100,10)),
//           DCB=(RECFM=VB,BLKSIZE=16096),
//           DISP=(MOD,PASS)
```

  For more information on C language processor JCL, see the "C Language Processor" chapter in the *Compuware Shared Services User/Reference Guide*.

  The above example is a temporary file. You can save the JCL as a permanent file to be reused as shown in the example below:

```
//SYSPRINT DD DSN=CX,STORED.LISTINGS(MEMBER)
```

- The SYSPRINT file will be LRECL=133, RECFM=FBA.
- Add SLCXCNTL sample library member CXC as the postprocessor step following your C compile step. CXC can be customized to meet your site's requirements.

  Note: The CWPLOAD file must be allocated using DISP=OLD or DISP=SHR.

- The object module that is processed by the postprocessor must be used as input to the linkedit step.

See the *Compuware Shared Services User/Reference Guide* for more information on this JCL and C language processor options.

Step 10. Continue with ECC Customization

If you have any Compuware products that have been licensed for use by your organization, you must continue with the procedures in **Chapter 6, "LMS Customization"**.

Chapter 6.
LMS Customization

This chapter provides the basic procedures for setting up and customizing Compuware's License Management System (LMS) for use with the Compuware products licensed by your organization. It includes instructions for:

- Transferring a License Certificate to your host system.
- Setting up and using the License Administration Utility (LAU) to create a License File and define system nodes.
- Importing a Certificate into a License File.
- Initializing the LMS runtime environment.

Preliminary Considerations

Which tasks you need to perform will depend on whether LMS has already been enabled at your site.

Running More than One License Management System

Under certain circumstances, a site may decide to run more than one instance of LMS on an MVS system. For instance, you might want both test and production LMS environments on the same MVS system. To make the production LMS environment the default LMS environment for the MVS system, designate it in the LMSINIT JCL and startup proc described in "Step 7. Prepare an APF Authorized LMSINIT" on page 6-13. Designate a different name for the test LMS environment in a separate LMSINIT JCL and proc. Then add a DD statement, pointing to the test LMS subsystem, to the startup code of the Compuware products you want to utilize that environment. To use different License Certificates in the test LMS environment, create a separate test LMS License File and use it as input to the LMSINIT JCL and proc for your LMS test environment. For more information on multiple LMS environments, see the chapter entitled "Creating Runtime Environment" in the *License Management System User/Reference Guide*.

Managing More than One Compuware Customer Number

A site that manages multiple Compuware customer numbers, for example a service bureau, will have a License File for each Compuware customer number. All of the relevant License Files for a system can be loaded into a single LMS environment by including the datasets as input when preparing the LMSINIT JCL and proc as described in "Step 7. Prepare an APF Authorized LMSINIT" on page 6-13, then running LMSINIT as described in "Step 8. Run LMSINIT" on page 6-21. By using the file access security set up by the service bureau for their customers, LMS provides control over which customers can use which License File. For more information, refer to the appendix entitled "Service Bureau Environment" in the *License Management System User/Reference Guide*.

Note: Any use of Compuware products by third parties must be authorized in your License Agreement.

Which Steps Do I Need to Perform?

The choice of which steps to perform depends on whether LMS has already been installed and initialized as part a previous Compuware product installation. For this reason, the procedures in this chapter are tailored for two possible scenarios:

- You need to install and set up LMS for the first time, then enable it to work with the Compuware product being installed.
- You are an established LMS user who only needs to enable it to work with the Compuware product being installed.

As shown in Table 6-1, a first-time user must perform every step. Most established LMS users can skip Steps 3, 4, and 5.

Table 6-1. LMS Steps Required

First-Time User	Step	Established User
x	"Step 1. Transfer License Certificate to Host"	x
x	"Step 2. Set up License Administration Utility (LAU)"	x
x	"Step 3. Create License File" on page 6-6	
x	"Step 4. Verify License File" on page 6-8	x
x	"Step 5. Define Nodes" on page 6-9	
x	"Step 6. Import License Certificate" on page 6-11	x
x	"Step 7. Prepare an APF Authorized LMSINIT" on page 6-13	x
x	"Step 8. Run LMSINIT" on page 6-21	x
x	"Step 9. Verify Proper LMS Function" on page 6-22	x
x	"Step 10. Continue ECC Customization" on page 6-23	x

Installing and Enabling LMS

Follow the instructions below that apply to your situation.

Step 1. Transfer License Certificate to Host

LMS uses License Certificate files to configure access to Compuware products licensed by your organization. A License Certificate is a text file typically sent to your site via e-mail by Compuware's Worldwide License Management team. To be used by LMS, a License Certificate must be accessible to MVS.

1. Locate the License Certificate for the Compuware products being installed. A License Certificate file can contain licensing for more than one Compuware product.
2. Allocate a target dataset on the host system for the License Certificate. Use the DCB parameters RECFM=FB, LRECL=80, and whatever BLKSIZE you prefer.
3. Transfer the License Certificate to the host using File Transfer Protocol (FTP), IND$FILE, cut and paste, or any other method desired.

   Note: If you open the License Certificate in an ISPF editor or transfer it using cut and paste, make sure NUMBERS is set to OFF.

4. If you have more than one License Certificate file, repeat the process as required.

Step 2. Set up License Administration Utility (LAU)

This step creates the CLIST for starting the LAU, prepares the LAU to run, and makes it accessible to the License Administrators at your site.

1. Enter the following command to execute the LMSPREP EXEC from the SLMSCNTL library.

   ```
   TSO EX ' your SMP/E dataset prefix.SLMSCNTL(LMSPREP)'
   ```

Figure 6-1. Prepare License Administration Utility Screen

```
---------------------- Compuware License Management 02.00.00 -------------------

                      Prepare License Administration Utility




    Type the data set prefix that will be used to execute the License
    Administration Utility.  The prefix shown below matches the
    prefix that you selected for your License Management SMP/E data
    sets.  If you plan to execute the License Management System from
    data sets other than the License Management SMP/E target or
    distribution libraries, then enter the appropriate data set
    prefix below.

            Data Set Prefix. LM.PROD.RO20000
```

The following messages will notify you that you have successfully tailored the SLMSCNTL screen.

```
LMSP100I - SLMSCNTL(CWLMA) SUCCESSFULLY TAILORED
LMSP100I - SLMSSENU(LMASTPLB) SUCCESSFULLY TAILORED
LMSP100I - SLMSCNTL(CWLMQURY) SUCCESSFULLY TAILORED
LMSP100I - SLMSCNTL(CWLMDMSG) SUCCESSFULLY TAILORED
LMSP100I - SLMSCNTL(CWLMCHKP) SUCCESSFULLY TAILORED
```

2. The second screen will be displayed only when the dataset prefix field displayed in the first screen was changed to a value that did not match the dataset prefix of the SLMSCNTL library that contained LMSPREP. This is to accommodate customers who do not want to execute the LAU directly from the SMP/E target libraries for the License Management System.

Figure 6-2. Prepare License Administration Utility - second screen

```
---------------------- Compuware License Management 02.00.00 -------------------

                      Prepare License Administration Utility




    Please copy the contents of the SLMSCNTL and SLMSSENU SMP/E data sets
    to the non-SMP/E controlled data sets that you wish to execute the
    License Management System from.
```

3. Perform either of the following:
   - Add the SLMSCNTL library to your SYSPROC concatenation, or
   - Copy member CWLMA from the SLMSCNTL library into a dataset already concatenated to your SYSPROC DD.
4. Add the following line to an ISPF selection panel of your choice:

   ```
   L,'CMD(%CWLMA) NOCHECK'
   ```

   Note: Adding CWLMA to ISPF is optional, and you may execute CWLMA as a CLIST.

5. If your site has elected to secure the ISPF Command Table via an external security manager (i.e., RACF, TOPSECRET, or ACF2), then you must take appropriate steps to identify the command LMAMAIN to your external security manager.
6. If you are an established user, go to “Step 4. Verify License File” on page 6-8.

Step 3. Create License File

If you have previously installed Compuware's License Management System software, go to "Step 4. Verify License File" on page 6-8.

In this step, you will use the LAU to create a License File.

1. Start the LAU by selecting the ISPF menu item that was added as part of "Step 2. Set up License Administration Utility (LAU)" on page 6-4. The License File Selection screen (Figure 6-3) will be displayed.

Figure 6-3. License File Selection Screen

```
 ---------------------- Compuware License Management 02.00.00 -- Row 1 to 2 of 2
Command ===>                                                  SCROLL ===> PAGE
                            License File Selection

Current Selection:
Enter New DSN  . .
                 (fully qualified without quotes)
   Delete/Define . N (Y|N)

 OR select below:-

Action     DSN                                              Added by

****************************** Bottom of data ********************************
```

2. Type the name you want to use for your License File (fully qualified without quotes) in the Enter New DSN field.
3. Type **Y** in the Delete/Define field.
4. Press Enter. The Delete/Define and Initialize License File screen (Figure 6-4) will be displayed, and the file name you specified above will be shown in the New License File field.

Figure 6-4. Delete/Define and Initialize License File Screen

```
---------------------- Compuware License Management 02.00.00 -- Row 1 to 2 of 2
Command ===>                                                  SCROLL ===> PAGE
                    Delete/Define and Initialize License File

New License File . : TSOID01.License.File.New

 Unit . . . . . . . . _________   (required for JES3 only)
Volume Serial  . . . _______     (blank for system determined volume)

Edit JCL . . . . . . Y  (Y-Yes,N-No)

     Press END Key to skip this process

Jobcard:
//Job Card information line 1
//Job Card information line 2
//Job Card information line 3
//Job Card information line 4

Select     Node      Description

****************************** Bottom of data ********************************
```

5. Type **Y** in the Edit JCL field.
6. Type the information for your jobcard in the four lines of the Jobcard field.

Note: If you are not using all four lines of the job card, comment out the unused lines.

7. Press Enter. The Edit Delete/Define JCL screen (Figure 6-5) will be displayed.

Figure 6-5. Edit Delete/Define JCL Screen

```
 File Edit Confirm Menu Utilities Compilers Test Help
------------------------------------------------------------------------------
EDIT       SYS98274.T095731.RA000.TSOID01.R0111344            Columns0000100080
Command ===>                                                  Scroll ===>CSR
************************************ Top of Data ***************************
000001 //Job Card information line 1
000002 //Job Card information line 2
000003 //Job Card information line 3
000004 //Job Card information line 4
000005 //*
000006 //*********************************************************************
000007 //* DELETE/DEFINE COMPUWARE LICENSE FILE
000008 //*
000009 //* JCL GENERATED BY TSOID01 ON 2001-05-17 AT 15:22
000010 //*********************************************************************
000011 //*
000012 //LFDELDEF EXEC PGM=IDCAMS
000013 //*
000014 //SYSPRINT DD SYSOUT=*
000015 //SYSIN DD *
000016 DELETE (TSOID01.License.File.New)
000017 SET MAXCC = 0
000018 DEF CL(NAME(TSOID01.License.File.New)
000019       IXD
```

8. Make any necessary edits, then enter the END command. A Confirm Submission window will be displayed over the Delete/Define and Initialize License File screen.
9. Type **Y** in the Confirm Submission window and press Enter. The JCL to create your License File will be submitted.
10. Enter the END command. The License File Selection screen (Figure 6-3) will be displayed.
11. After your job has completed with a good return code, type **S** in the Action field next to your new License File and press Enter. Your License File name will appear in the Current Selection field.
12. Enter the END command.
    - If you are a first-time user, the main Compuware License Management screen (Figure 6-8 on page 6-9) will appear.
    - If you are an established user, the Parameter Option screen (Figure 6-9 on page 6-9) will appear. To access the main Compuware License Management screen (Figure 6-8 on page 6-9), enter the END command.

Step 4. Verify License File

In this step, you will use the LAU to verify which License File you will be using.

1. Start the LAU by selecting the ISPF menu item that was added as part of "Step 2. Set up License Administration Utility (LAU)" on page 6-4. Because the LAU has been run before and a License File already exists, the main Compuware License Management screen shown in Figure 6-8 on page 6-9 will be displayed. If the dataset name of the License file that you want to use is displayed next to the Browse option, go to "Step 5. Define Nodes" on page 6-9.
2. Type 0 in the Option field and press Enter. The Parameter Option screen (Figure 6-6) will be displayed.

Figure 6-6. Parameter Option Screen

```
 ---------------------- Compuware License Management 02.00.00 ------------------
Option ===>

 1)   Select      License File - TSOID01.LICENSE.FILE
 2)   Node        Specify System Nodes




 X) Exit          Return to previous panel




      (C) Copyright 1998, Compuware Corp. All Rights Reserved.
```

3. Type 1 in the Option field and press Enter. The License File Selection screen (Figure 5-7) will be displayed.

Figure 6-7. License File Selection Screen

```
 ---------------------- Compuware License Management 02.00.00 -- Row 1 to 2 of 2
Command ===>                                                  SCROLL ===> PAGE
                            License File Selection

Current Selection:
Enter New DSN  . .
                (fully qualified without quotes)
   Delete/Define . N (Y|N)

OR select below:-

Action     DSN                                          Added by
  _        LICENSE.FILE1                                USERID1  2000-08-07
  _        LICENSE.FILE2                                USERID1  2000-08-07
****************************** Bottom of data ********************************
```

4. Type **S** in the Action field next to the file that you want to use and press Enter. Your selected file should appear in the current selection field.

Step 5. Define Nodes

If your installation does not consist of multiple nodes in a JES network, or if you do not require the ability to route License Management work (Import, Export, or Reporting) from one JES node to another in your network, go to “Step 6. Import License Certificate” on page 6-11.

In this step, you will use the LAU to define (to LMS) the network nodes on which License Management maintenance or reporting tasks could be dispatched.

Figure 6-8. Main Compuware License Management Screen

```
 ---------------------- Compuware License Management 02.00.00 ------------------
Option ===>

 0)    Parameters Specify System Parameters
 1)    Browse     License File - TSOID01.License.File
 2)    Update     License File
 3)    IMPORT     Import License Certificate
 4)    EXPORT     Export License Certificate
 5)    Reports    Run Reports

 6)    Disaster   Enable Disaster Site
 7)    Emergency  Emergency Password

 X)    Exit License Management




        Copyright (c) 1999 Compuware Corporation. All Rights Reserved.
Unpublished rights reserved under the Copyright Laws of the United States.
          Enter HELP for Copyright/Trade Secret Notice information.
```

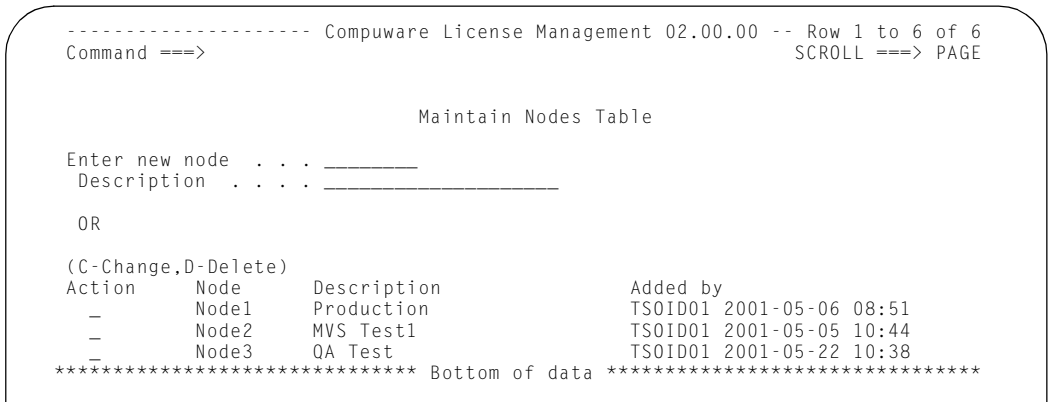
1. Type **0** in the Option field and press Enter. The Parameter Option screen (Figure 6-9) will be displayed.

Figure 6-9. Parameter Option Screen

```
 ---------------------- Compuware License Management 02.00.00 ------------------
Option ===>

 1)   Select      License File - TSOID01.LICENSE.FILE
 2)   Node        Specify System Nodes




 X) Exit          Return to previous panel




      (C) Copyright 1998, Compuware Corp. All Rights Reserved.
```

2. Type **2** in the Option field and press Enter. The Maintain Nodes Table screen (Figure 6-10) will be displayed.

Figure 6-10. Maintain Nodes Table Screen



3. In the Enter new node field on the Maintain Nodes Table (Figure 6-10), specify the name of the network node you want to add to the display list. This name identifies your installation's local JES in a network of systems or system complexes being used for network job entry (NJE) tasks. The node name can be up to eight alphanumeric characters.
4. Describe the new node briefly in the Description field. This description can be up to 20 alphanumeric characters.
5. Press Enter. The screen will be redisplayed with the new node in the selection list at the bottom of the screen. Repeat tasks step 3 through step 5 until you have defined all of the required nodes.
6. Enter the END command. The Parameter Option screen (Figure 6-9) will be displayed.
7. Enter the END command again. The main Compuware License Management screen (Figure 6-8) will be displayed.

Step 6. Import License Certificate

In this step the License Certificate for the Compuware product being installed will be imported into your site's License File.

1. On the main Compuware License Management screen (Figure 6-8 on page 6-9), type **3** in the Option field and press Enter. The first of two Import License Certificate screens (Figure 6-11) will be displayed.

Figure 6-11. First Import License Certificate Screen

```
---------------------- Compuware License Management 02.00.00 ------------
 Command ===>

                            IMPORT License Certificate

 Process DSN  . : TSOID01.License.File

 IMPORT From  . . LICENSE.CERTIF_____________________
                                          (fully qualified without quotes)
 Preview Only . . N (Y|N)
```

2. Type the name of the dataset used for your License Certificate file (fully qualified without quotes) in the IMPORT From field.
3. Type **N** in the Preview Only field.

   Note: If you prefer to generate and review a report detailing the implications of the License Certificate importation, enter Y in the Preview Only field. You then need to repeat this step with Preview Only set to N.

4. Press Enter. The second Import License Certificate screen (Figure 6-12) will be displayed.

Figure 6-12. Second Import License Certificate Screen

```
---------------------- Compuware License Management 02.00.00 -- Row 1 to 5 of 6
 Command ===>                                                 SCROLL ===> PAGE

                            IMPORT License Certificate

 Process DSN  . : TSOID01.License.File
 IMPORT From  . : LICENSE.CERTIF
 Preview Only . : N


 Edit JCL . . . . Y (Y-Yes,N-No)

 Jobcard:
 //Job Card information line 1
 //Job Card information line 2
 //Job Card information line 3
 //Job Card information line 4

 Select     Node      Description
   _        Node1     Production
   _        Node2     MVS Test1
   _        Node3     QA Test
****************************** Bottom of data ***********************
```

5. Type **Y** in the Edit JCL field.
6. Type the information for your job card in the four lines of the Jobcard field.

Note: If you are not using all four lines of the job card, comment out the unused lines.

7. Type **S** next to the nodes on which you want the Import job to run. If no node is selected, the job is submitted without any specific routing.

   Note: If you are running on only one node, do not make a selection here.

8. Press Enter. The Edit JCL screen (Figure 6-13) will be displayed.

Figure 6-13. Edit JCL Screen

```
 File Edit Confirm Menu Utilities Compilers Test Help
--------------------------------------------------------------------------------
EDIT SYS98274.T095731.RA000.TSOID01.R0111344              Columns0000100080
Command ===>                                                Scroll ===>CSR
********************************** Top of Data ******************************
000001 //Job Card information line 1
000002 //Job Card information line 2
000003 //Job Card information line 3
000004 //Job Card information line 4
000005//*
000006//*********************************************************************
000007//* IMPORT COMPUWARE LICENSE CERTIFICATE
000008//*
000009//* JCL GENERATED BY TSOID01 ON 2001-07-08 AT 15:32:17
000010//*********************************************************************
000011//*
000012//IMPORT EXEC PGM=LMALFIM,
000013// PARM='UPDATE'
000014//STEPLIB DD DSN=LM.DEVL.LOAD,
000015// DISP=SHR
000016//*
000017//CWLF0000 DD DSN=TSOID01.License.File,
000018// DISP=SHR
000019//CWLFIMP DD DSN=LICENSE.CERTIF,
```

9. Make any necessary edits, then enter the END command. A Confirm Submission window will be displayed over the second Import License Certificate screen.
10. Type **Y** in the Confirm Submission window and press Enter. The JCL to create your License File will be submitted.
11. Enter the END command. The main Compuware License Management screen (Figure 6-8 on page 6-9) will be displayed.
12. Type **X** in the Option field and press Enter to exit the LAU.
13. Proceed as follows:
    - If you are a first-time user, continue with “Step 7. Prepare an APF Authorized LMSINIT”.
    - If you are an established user, go to “Step 8. Run LMSINIT” on page 6-21.

Step 7. Prepare an APF Authorized LMSINIT

If you are an established user, please ensure that the modules listed below are copied from your License Management SMP/E target load library into your License Management APF authorized load library and go to “Step 8. Run LMSINIT” on page 6-21.

In this step, you will ensure that the LMS modules will be loaded from an APF authorized library, and you will also create the customized proc, JCL, and parameters dataset that will be used by LMSINIT to build the LMS runtime environment.

1. In order to function correctly, the LMS load modules must reside in an MVS APF authorized library. Consult your site’s MVS system programmer for details. This can be accomplished in either of the following ways:
   - APF authorize the library in which all the load modules for LMSINIT currently reside.
   - Move the following LMSINIT dataset members into an existing APF-authorized library:
     - LMACG000
     - LMADSALC
     - LMADTCV
     - LMALFAC
     - LMSAESTA
     - LMASFAC
     - LMSASSYS
     - LMSCHKPT
     - LMSCPUID
     - LMSDCTLG
     - LMSDIDCM
     - LMSDMAIN
     - LMSDSV99
     - LMSEMERP
     - LMSENMGR
     - LMSGETMS
     - LMSICRET
     - LMSIDLET
     - LMSIDUMP
     - LMSIESTA
     - LMSIFRR
     - LMSILAKC
     - LMSIMAIN
     - LMSINIT (link edited with AC=1)
     - LMSIPARM
     - LMSIPRNT
     - LMSIREAD
     - LMSISMF
     - LMSISRB
     - LMSIUPDT
     - LMSMSGEN
     - LMSMSGLT
     - LMSPCARR
     - LMSPCRTN
     - LMSRACF
     - LMSRSMGR
     - LMSSCOMM
     - LMSSDUMP
     - LMSSERVR
     - LMSSMF
     - LMSTRACE

2. Parameters used by LMSINIT are read from a SYSIN dataset. Edit the sample dataset provided in SLMSCNTL(LMINPARM) (see Figure 6-14).

   – Multiple operand identifiers and their associated values are allowed on a single record in the input file.

   – The operand and its associated value can start in any column and can be continued on more than one statement.

   – Columns 1 through 71 are available for use.

   – Column 72 is reserved.

   – Columns 73 through 80 can be used for sequence numbers. They are ignored by LMSINIT.

   – To continue an operand on a second statement, complete the first statement through column 71, and continue the operand on the next statement starting in column 1.

   – Only the value specified for SUBSYSTEM_ID is case-sensitive. All other operands and their values are folded to upper case by LMSINIT.

   – Only one occurrence of each of the operands is allowed. If duplicates exist, the last one encountered will be used.

   – An asterisk '*' in column 1 causes the entire statement to be a comment.

   – Line level comments are supported using the '/*' to start a comment and '*/' to end the comment. Imbedded comments are supported.

   For more information about specific parameters, refer to the chapter entitled "Creating Runtime Environment" in the *License Management System User/Reference Guide*.

   Note: The following parameters are required: FUNCTION(UPDATE), SITE(), SUBSYSTEM_ID(), CHKPT_DSNAME(). CHKPT_DSNAME() is new with release 2.0 of LMS.

Figure 6-14. Sample LMSINIT Parameters Dataset (Part 1 of 6)

```
*****************************************************************
**   LMINPARM - SAMPLE LMSINIT PARAMETER FILE                  **
*****************************************************************
*
*****************************************************************
**                                                             **
**    The following are REQUIRED parameters for LMSINIT        **
**    execution, preceded by a brief description of valid      **
**    entries.                                                 **
**                                                             **
**    Consult the ECC: LMS USER REFERENCE GUIDE for a          **
**    more detailed description of these LMSINIT parameters    **
*****************************************************************
*****************************************************************
**    UPDATE - Creates a new LMS Subsystem and/or
**             updates an existing subsystem
**    CREATE - Creates a new LMS Subsystem
**    DELETE - Deletes an existing LMS Subsystem
*****************************************************************
*
   FUNCTION(UPDATE)          /* CREATE, UPDATE, OR DELETE */
*
*****************************************************************
**   This number (or numbers) tell LMSINIT which site(s) to
**   load from the license file(s) into the license cache.
**   Site numbers are defined in the license certificates.
**
**   There are three different formats of the SITE operand.
**
**   1. SITE(nnn)
**   2. SITE(ddname,nnn)
**   3. SITE((ddname1,nnn),(ddname2,mmm).....,(ddnamen,ppp))
*****************************************************************
*
   SITE()                    /* 3 DIGIT SITE NUMBER       */
*
*****************************************************************
**    The value specified on the SUBSYSTEM_ID operand names
**    the OS/390 subsystem that this invocation of LMSINIT
**    is to process. This value can contain upper and lower
**    case letters and numbers, and any special characters.
**
**    It is necessary to coordinate the use of subsystem
**    identifier, such that no duplicate names exist,
**    either within the set of License Management
**    subsystems, or across all subsystems defined to OS/390.
**    You should check with your systems programmer to ensure
**    that the SUBSYSTEM_ID value you have chosen is not
**    already in use.
*****************************************************************
*
   SUBSYSTEM_ID()            /* 4 CHARACTER SUBSYSTEM ID  */
*
*****************************************************************
**    This operand defines whether this SUBSYSTEM_ID is to
**    be declared the default subsystem DEFAULT(YES),
**    DEFAULT(YES,FORCE) or is not DEFAULT(NO).
**
**    Default Subsystems require special authority to define.
**
**    See "The Default Subsystem" topic in the ECC: LMS USER
**    REFERENCE GUIDE for a discussion of how this parameter
**    affects license processing.
*****************************************************************
```

Figure 6-15. Sample LMSINIT Parameters Dataset (Part 2 of 6)

```
*
   DEFAULT()               /* "NO" "YES" OR "YES,FORCE" */
*
*****************************************************************
**
**    LMS 2.0 requires a checkpoint data set be available to
**    LMSINIT and to certain operating system exits at all
**    times. This data set is created automatically by LMINIT
**    if it does not already exist, or it is updated if it does
**    exist.  You must specify the name of this data set,
**    either using the parameters described here, or by a
**    special DD statement in the LMSINIT JCL.
**
*****************************************************************
*
*****************************************************************
*
**    CHKPT_DSNAME specifies the 1 to 44 character dsname of
**    the LMS 2.0 checkpoint data set. LMSINIT will create this
**    data set automatically, if it does not already exist.
**    NOTE: The USERID under which LMSINIT is run MUST have
**    ALTER access to the security system (RACF, ACF/2,
**    TOPSECRET) entity named by this DSNAME. The IDCAMS
**    utility is dynamically invoked to define this data set.
**    See the other (optional) checkpoint data set parameters
**    later in this member.
**
**
*****************************************************************
*
   CHKPT_DSNAME()
*
*****************************************************************
**    The following are OPTIONAL parameters for LMSINIT      **
**    execution, preceded by a brief description of valid    **
**    entries.                                               **
**                                                           **
**    Consult the ECC: LMS USER/REFERENCE GUIDE for a more   **
**    detailed description of these LMSINIT parameters.      **
*****************************************************************
*
*****************************************************************
**    The SERVICE_BUREAU operand controls the use of an      **
**    external security manager (RACF, ACF/2, or TOPSECRET)  **
**    in determining a user's access to licenses in the      **
**    cache.  This operand is used ONLY for invocations      **
**    of LMSINIT that contain multiple //CWLFxxxx DD         **
**    statements or specifies multiple SITE values.          **
*****************************************************************
*
   SERVICE_BUREAU()        /* "YES" OR "NO" */
*****************************************************************
**    The value specified on the SMF_ID operand defines the
**    SMF Record ID number that is to be used on all SMF
**    records written by the License Management System.
**    The SMF_ID must be a 3 digit number in the range of
**    128 through 255.
*****************************************************************
*
   SMF_ID()                /* 1-3 DIGIT SMF RECORD ID   */
*
*****************************************************************
**    The value specified on the GTF_ID operand defines the
**    GTF Record ID number that is to be used on all GTF
**    records written by the License Management System.
**
**    Note:  You should not specify a GTF_ID unless your are
**    diagnosing a license management program error and are
**    directed by Compuware to invoke GTF recording.
*****************************************************************
```

Figure 6-16. Sample LMSINIT Parameters Dataset (Part 3 of 6)

```
*
   GTF_ID()                 /* 1-5 DIGIT GTF RECORD ID   */
*
*****************************************************************
**    The value specified on the LANGUAGE operand specifies
**    the language that error messages are to use.  This two
**    character operand can specify the following languages:
**
**    EN - English
**
**    If LANGUAGE is not specified,
**    then 'EN' (English) is chosen.
**
*****************************************************************
*
   LANGUAGE()               /* 2 CHAR COUNTRY CODE       */
*
*****************************************************************
**    LMSINIT has the ability to load an emergency license
**    cache even if the license file is unavailable or has
**    become corrupted.
**
**    The value required on the EMERGENCY operand is obtained
**    by calling the Worldwide Licence Management department
**    at Compuware, and by requesting an emergency password.
**    This password can also be entered using the License
**    Administration Utility to update the license file.
**
**    If, however, the license file or LAU is not available
**    you can specify the EMERGENCY password in the LMSINIT
**    parameter dataset.  Then remove all CWLFnnnn DD
**    statements from the LMSINIT execution JCL and submit
**    the JCL.
**
**    When a legitimate emergency password is present in the
**    LMSINIT sysin dataset, all Compuware products will be
**    allowed to execute on the CPUs that LMSINIT established
**    a license cache for.
**
**    The password includes its expiration date.
**    The expiration date can be from 2 to 14 days from the
**    date of issue.
*****************************************************************
*
   EMERGENCY()              /* EMERGENCY PASSWORD        */
*
*****************************************************************
*
**    CHKPT_VOLSER specifies the 6 character volume serial of
**    the DASD volume on which the checkpoint data set is to
**    reside.  This parameter is optional. If it does not exist,
**    your system installation defaults for VSAM data sets will
**    be used to determine the placement of this data set. You
**    must insure that any VOLSER your specify is consistent
**    the SMS class definitions that may also exist.
**
*****************************************************************
*
   CHKPT_VOLSER()
*
*****************************************************************
*
**    CHKPT_STORCLASS, CHKPT_DATACLASS and CHKPT_MGMTCLASS
**    specifies the names your installation has chosen to
**    describe the allocation of this VSAM checkpoint data set.
**    These parameters are optional, but if specified, will be
**    used within the IDCAMS DEFINE control statements when
**    the data set is created.
**
*****************************************************************
```

Figure 6-17. Sample LMSINIT Parameters Dataset (Part 4 of 6)

```
*
   CHKPT_STORCLASS()
   CHKPT_DATACLASS()
   CHKPT_MGMTCLASS()
*
*****************************************************************
*
**     The value specified on the SITE_WARNING parameter
**     specifies whether you want LMSINIT to complete with a
**     return code of 4 whenever SITE records in a license file
**     are skipped (YES), or you want LMSINIT to complete with
**     a return code of 0 (NO). LMS 1.0 always completed with a
**     return code of 4 when LMSINIT detected that there were
**     SITE records in the license file that were not loaded,
**     because their SITE number was not included in the SITE()
**     parameter. If you omit the SITE_WARNING parameter, this
**     behavior will exist in LMS 2.0 as well.  But if you know
**     that you are skipping SITE records, and you want a
**     return code of 0 so that your automated operator program
**     will detect this return code, then include the
**     SITE_WARNING parameter and specify a value of NO.  If
**     this parameter is omitted SITE_WARNING(YES) is used as a
**     default.
**
*
*****************************************************************
*
   SITE_WARNING()          /* YES or NO                */
*
*****************************************************************
*
**     The next three LMSINIT parameters define master console
**     commands which can automatically be issued by LMSINIT.
**     LMSINIT can complete with a return code of 0, a return
**     code of 4 or a return code of 8 or greater. Each of
**     these three conditions can have a unique console command
**     associated with it. LMSINIT will issue the command that
**     represents the current return code.
**
**     Compuware does not supply any procedures to be started
**     by these commands. It is the customer's responsibility
**     to insure that any PROC specified in a command exists in
**     an appropriate procedure library.
**
**     The examples below issue an OS/390 "START" command. But
**     any valid operator command could be issued as well.
**
**     If no command is specified for a particular return code
**     value, then no command is issued when that return code
**     occurs.
**
**     Any, all or none of the three return code conditions
**     can have a command associated with it.
**
**     Commands can be the same or different for each of the
**     three return codes.
**
**     The commands will be issued as if they were entered at
**     the Master Console, and the Security (RACF, ACF/2 or
**     TOPSECRET) USERID will be the USERID under which LMSINIT
**     is running. Insure that this USERID has the appropriate
**     authority to issue Master Console commands.
**
**     Rules for coding these commands follow:
**
**     If the command contains blanks, then enclose the entire
**     command in single (') or double (") quotes.
**
**     If the command contains blanks and single quotes, then
**     enclose the entire command in double (") quotes.
**
**     If the command contains blanks and double quotes, then
**     enclose the entire command in single (') quotes.
**
**     The single or double quotes will be removed before the
**     command is issued.
```

Figure 6-18. Sample LMSINIT Parameters Dataset (Part 5 of 6)

```
**
**     Example:
**
**     SUCCESS_CMD("START someproc,PARM='RC=0'")
**     WARNING_CMD('START othrproc,PARM="RC=4"')
*
*****************************************************************
*
   SUCCESS_CMD()           /* ISSUED IF RETURN CODE = 0 */
   WARNING_CMD()           /* ISSUED IF RETURN CODE = 4 */
   ERROR_CMD()             /* ISSUED IF RETURN CODE > 4 */
*
*****************************************************************
**
**    The remaining LMSINIT parameters define the E-mail
**    Notification Facility (ENF).
**
**    If you are not using this facility, you may skip the
**    rest of these parameters.
**
*****************************************************************
*
*****************************************************************
**    TCPIP_NAME specifies the name of the TCP/IP protocol
**    stack that is active on this CPU
*****************************************************************
*
   TCPIP_NAME()             /* NAME OF TCP/IP REGION     */
*
*****************************************************************
**    The EMAIL() parameter further limits the number of E-MAIL
**    messages that are automatically generated.   The defaults
**    are shown below. The values that can be coded in place
**    of "WARN" and "NONE" or "FAIL". "NONE" specifies that no
**    E-MAIL messages are to be generated for the product or
**    the option. "FAIL" specifies that only E-MAIL messages
**    reporting failure conditions (i.e. the product or the
**    option is not allowed to execute) are to be generated.
**    "WARN" specifies that both warning and failure messages
**    are to be generated.  E-MAIL messages are still only
**    generated once per product/option per CPU per day, but
**    by specifying the EMAIL parameter, even these messages
**    can be further limited.  This parameter does not affect
**    the messages that products display when warning or
**    failure conditions occur. This parameter ONLY limits the
**    generation of E-MAIL messages.
*****************************************************************
*
   EMAIL(PRODUCT(WARN),OPTION(WARN))  /* limit E-MAIL    */
*
*****************************************************************
**    For Release 3.4 and above of IBM's TCP/IP only:
**    Specify a USERID that the License Management Run Time
**    Environment is to use.
*****************************************************************
*
   TCPIP_USERID()           /* TCP/IP USERID             */
*
*****************************************************************
**    Specify the name, as defined in the domain name server
**    used by this mainframes TCP/IP, of the gateway host
**    used in your network to access the Internet.
**
**    Note: If you specify INTERNET_GATEWAY_NAME, you cannot
**    specify INTERNET_GATEWAY_ADDR, These two operands are
**    alternate methods for specifying the same resource
**    to LMSINIT. One of these two methods MUST be chosen,
**    but both of them cannot be.
*****************************************************************
```

Figure 6-19. Sample LMSINIT Parameters Dataset (Part 6 of 6)

```
*
   INTERNET_GATEWAY_NAME()  /* NAME OF INTERNET GATEWAY  */
*
*************************************************************
**    Specify the address (in dotted decimal notation of the
**    form nnn.nnn.nnn.nnn, where n is a decimal number from
**    1 to 255) of the gateway host used in your network to
**    access the Internet.
**
**    Note: If you specify INTERNET_GATEWAY_ADDR, you cannot
**    specify INTERNET_GATEWAY_NAME. These two operands are
**    alternate methods for specifying the same resource
**    to LMSINIT. One of these two methods MUST be chosen,
**    but both of them cannot be.
*************************************************************
*
   INTERNET_GATEWAY_ADDR()  /* ADDR OF INTERNET GATEWAY  */
*
*************************************************************
**    Specify the port number on your Internet gateway host
**    that is used for SMTP traffic destined for the Internet.
**    The default port number that LMSINIT assigns if this
**    operand is omitted is 25, and in all but a very few
**    cases this default is appropriate.
*************************************************************
*
   INTERNET_GATEWAY_PORT()  /* PORT NUMBER ON GATEWAY    */
*
*************************************************************
**    Specify the Internet e-mail address (or name) in the
**    form name@institution.type of the individual
**    (or department) that is to be designated as the sender
**    of e-mail messages. This name will appear as the
**    FROM: name on all e-mails automatically generated by
**    the License Management System.
*************************************************************
*
   MAIL_FROM_NAME()         /* INTERNET E-MAIL FROM NAME */
*
*************************************************************
**    Specify the Internet e-mail address (or name) in the
**    form name@institution.type of the individual
**    (or department) that is to receive all automatically
**    generated e-mail messages relating to product
**    licensing errors.
*************************************************************
*
   MAIL_TO_SEC_NAME()       /* INTERNET E-MAIL TO NAME   */
*
*************************************************************
**    Specify the Internet e-mail address (or name) in the
**    form name@institution.type of the individual
**    (or department) that is to receive all automatically
**    generated e-mail messages relating to Compuware
**    License Management software errors (program ABENDs).
*************************************************************
*
   MAIL_TO_ABN_NAME()       /* INTERNET E-MAIL TO NAME   */
*
*
* End of Parameter listing
*
```

Note: For the MAIL_TO_ABN_NAME() parameter, entering an Internet address instructs LMS to send an e-mail to this address for every LMS abend encountered. For Domestic U.S. customers only, enter CSS@COMPUWARE.COM if you want to send these e-mails directly to Compuware License Management System product support.

3. To ensure access to Compuware products is automatically enabled, establish a procedure in your SYS1.PROCLIB to launch LMSINIT as a started task during IPL and IML processing. Consult your site's MVS system programmer for details. Edit the

sample proc provided in SLMSCNTL(LMINPROC) (Figure 6-20). Change the dataset names to match those used at your site.

Note: This proc must run before any other Compuware product procs.

Figure 6-20. Sample LMSINIT Proc

```
//LMSINIT PROC
//LMSSTEP EXEC PGM=LMSINIT,PARM='LANGUAGE=EN'
//STEPLIB   DD DISP=SHR,DSN=USER.LMS.AUTHLOAD        <==== CHANGE DSN
//SYSPRINT  DD SYSOUT=*
//CWLF0000  DD DISP=SHR,DSN=YOUR.VSAM.LICENSE.FILE   <==== CHANGE DSN
//LMSCHKPT  DD DSN=<YOUR.DATA.SET.NAME>,             <==== CHANGE DSN
//             DISP=(MOD,KEEP,KEEP),
//             SPACE=(TRK,0),
//             UNIT=<UNIT>,                          <==== CHANGE UNIT
//             VOL=SER=<VOLSER>                      <==== CHANGE VOLSER
//SYSIN     DD DISP=SHR,
//             DSN=<SMPE.HLQ>.SLMSCNTL(LMINPARM)     <==== CHANGE DSN
```

4. Edit the sample JCL provided in SLMSCNTL(LMINJCL) (Figure 6-21). Change the dataset names to match those used at your site.

Figure 6-21. Sample LMSINIT JCL

```
//*
//* YOUR JOBCARD GOES HERE
//*
//LMSSTEP EXEC PGM=LMSINIT,PARM='LANGUAGE=EN'
//STEPLIB   DD DISP=SHR,DSN=USER.LMS.AUTHLOAD          <==== CHANGE DSN
//SYSPRINT  DD SYSOUT=*
//CWLF0000  DD DISP=SHR,DSN=YOUR.VSAM.LICENSE.FILE     <==== CHANGE DSN
//SYSIN     DD DISP=SHR,
//             DSN=<SMPE.HLQ>.SLMSCNTL(LMINPARM)       <==== CHANGE DSN
```

Step 8. Run LMSINIT

Submit your site's customized LMSINIT JCL created from the sample provided in SLMSCNTL(LMINJCL) (Figure 6-21). This will establish the runtime environment that enables LMS to function.

Step 9. Verify Proper LMS Function

In this step, you will use the LAU to generate a License Verification Report. You can then use that report to verify that the procedures in this chapter were performed correctly and LMS is operating properly.

1. Start the LAU by selecting the ISPF menu item that was added as part of "Step 2. Set up License Administration Utility (LAU)" on page 6-4. The main Compuware License Management screen shown in Figure 6-8 on page 6-9 will be displayed.
2. Type **5** in the Option field and press Enter. The Reports screen (Figure 6-22) will be displayed.

Figure 6-22. Reports Screen

```
---------------------- Compuware License Management 02.00.00 -- Row 1 to 3 of 7
Command ===>                                                  SCROLL ===> PAGE
                              Report Selection
Select "D" for Detail, or "S" for Summary
                         Expire Date   Process DSN
_ License Verification
_ Current Cache Report
_ VSAM License File
_ SAM License File
_ SMF License Records
_ Product Activity
_ Activity Extract

Edit JCL . . . . N
Jobcard:
//Job Card information line 1
//Job Card information line 2
//Job Card information line 3
//Job Card information line 4

Select     Node      Description
  _        Node1     Production
  _        Node2     MVS Test1
  _        Node3     QA Test
```

3. Type S next to License Verification.
4. Type **Y** in the Edit JCL field.
5. Type the information for your jobcard in the four lines of the Jobcard field.

   Note: If you are not using all four lines of the job card, comment out the unused lines.

6. Type **S** next to the nodes on which you want the Report job to run. If no node is selected, the job is submitted without any specific routing.

   Note: If you are running on only one node, do not make a selection here.

7. Press Enter. An Edit JCL screen will be displayed.
8. Make any necessary edits, then enter the END command. A Confirm Submission window will be displayed over the Reports screen.
9. Type **Y** in the Confirm Submission window and press Enter. The JCL to create your License Verification Report will be submitted.
10. Enter the END command. The main Compuware License Management screen (Figure 6-8 on page 6-9) will be displayed.
11. Type **X** in the Option field and press Enter to exit the LAU.

12. Use SDSF to examine the License Verification Report and ensure that the Compuware product being installed is recognized and enabled by LMS.

Step 10. Continue ECC Customization

If the Compuware product you are installing also requires customization of the Host Communications Interface (HCI), continue with **Chapter 7, "HCI Customization"**.

Note: This does not apply to Abend-AID or Abend-AID for CICS. Individuals installing these products have now completed the ECC installation and customization procedures. Please return to the appropriate product manual to continue with the procedure provided there.

Resolving Problems

If any error messages were returned while performing steps in this chapter, refer to the *Enterprise Common Components Messages and Codes* guide.

If you encountered any other difficulties related to License Management System, please consult the *License Management System User/Reference Guide* for more information. If problems persist, contact Compuware Technical Support.

Chapter 7.
HCI Customization

The procedures in this chapter are used to customize the Host Communications Interface (HCI).

CAUTION:
Abend-AID and Abend-AID for CICS users should not perform the steps in this chapter. Return to the appropriate Abend-AID installation and customization guide and continue with the product installation. Specific customization instructions for the HCI component are provided there.

UNIFACE or XPEDITER/DevEnterprise users must follow the instructions contained in this chapter.

Only two steps are required to customize HCI:

- Coding the HCICNFIG macros
- Assembling HCICNFIG

For more information about the HCI, please refer to **Chapter 8, "HCI Facilities"**.

Step 1. Coding HCICNFIG Macros

Structure of the HCICNFIG Assembly

Figure 7-1 shows a skeleton HCICNFIG assembly. Of the six HCI configuration macros, only HCICNGCA is required, but most installations will include the remaining five macros. They must be coded in the order shown. Any other order will be diagnosed as an error.

Figure 7-1. HCICNFIG SOURCE: Skeletal HCICNFIG Assembly (Page 1 of 3)

```
HCICNFIG TITLE '** HCICNFIG -- CONFIGURATION SPECIFICATIONS **'
*************************************************************************
**                                                                     **
**       HCICNFIG - THIS MODULE DEFINES THE CONFIGURATION              **
**                  MACROS USED FOR THIS EXECUTION OF THE HCI.         **
**                                                                     **
*************************************************************************
         SPACE 1
HCICNFIG CSECT ,
HCICNFIG AMODE 31
HCICNFIG RMODE ANY
*************************************************************************
**                                                                     **
**       SPECIFY GLOBAL HCI VALUES.                                    **
**                                                                     **
*************************************************************************
         HCICNGCA ,                                                    *
                  .
                  .      Operands for HCICNGCA macro
                  .
*************************************************************************
**                                                                     **
**       SPECIFY ACCESS METHOD BLOCK VALUES.                           **
**                                                                     **
*************************************************************************
         HCICNAMB ,                                                    *
                  .
                  .      Operands for HCICNAMB macro number one
                  .
         HCICNAMB ,                                                    *
                  .
                  .      Operands for HCICNAMB macro number two
                  .
         HCICNAMB ,                                                    *
                  .
                  .      Operands for HCICNAMB macro number 'n'
                  .
```

Figure 7-2. HCICNFIG SOURCE: Skeletal HCICNFIG Assembly (Page 2 of 3)

```
*********************************************************************
**                                                                 **
**      SPECIFY TCP/IP PORT DEFINITIONS.                           **
**                                                                 **
*********************************************************************
        HCICNPCB ,                                                  *
                 .
                 .      Operands for HCICNPCB macro number one
                 .
        HCICNPCB ,                                                  *
                 .
                 .      Operands for HCICNPCB macro number two
                 .
        HCICNPCB ,                                                  *
                 .
                 .      Operands for HCICNPCB macro number 'n'
                 .
*********************************************************************
**                                                                 **
**      SPECIFY JOURNAL DATA SETS.                                 **
**                                                                 **
*********************************************************************
        HCICNJCB ,                                                  *
                 .
                 .      Operands for HCICNJCB macro number one
                 .
        HCICNJCB ,                                                  *
                 .
                 .      Operands for HCICNJCB macro number two
                 .
        HCICNJCB ,                                                  *
                 .
                 .      Operands for HCICNJCB macro number 'n'
                 .
*********************************************************************
**                                                                 **
**      SPECIFY SIDE INFORMATION TABLE VALUES.                     **
**                                                                 **
*********************************************************************
        HCICNSIT ,                                                  *
                 .
                 .      Operands for HCICNSIT macro number one
                 .
        HCICNSIT ,                                                  *
                 .
                 .      Operands for HCICNSIT macro number two
                 .
        HCICNSIT ,                                                  *
                 .
                 .      Operands for HCICNSIT macro number 'n'
                 .
```

Figure 7-3. HCICNFIG SOURCE: Skeletal HCICNFIG Assembly (Page 3 of 3)

```
*************************************************************************
**                                                                     **
**      SPECIFY TP_PROFILE TABLE VALUES.                               **
**                                                                     **
*************************************************************************
        HCICNTPT ,                                                     *
                 .
                 . Operands for HCICNTPT macro number one
                 .
        HCICNTPT ,                                                     *
                 .
                 . Operands for HCICNTPT macro number two
                 .
        HCICNTPT ,                                                     *
                 .
                 . Operands for HCICNTPT macro number 'n'
                 .
*************************************************************************
**                                                                     **
**      SPECIFY END OF CONFIGURATION ASSEMBLY.                         **
**                                                                     **
*************************************************************************
        END HCICNFIG
```

Coding Rules

HCICNGCA Macro

There can be one, and only one, HCICNGCA macro, and it must be the first configuration macro specified.

HCICNAMB Macro

There can be zero or more HCICNAMB macros. For LU 2 and LU 6.2, each HCICNAMB macro defines a unique VTAM ACB, which must have a corresponding APPL statement in the VTAMLST dataset. For TCP/IP, each HCICNAMB defines a connection with a single TCP/IP address space. If an installation runs only one TCP/IP address space, then there would be only one HCICNAMB macro corresponding to the single TCP/IP. If an installation runs more than one TCP/IP address space, and if the HCI is to communicate with more than one TCP/IP at a time, then a separate HCICNAMB macro must be coded for each TCP/IP address space.

HCICNPCB Macro

There can be zero or more HCICNPCB macros. This macro applies to TCP/IP only, and is not used for LU 2 or LU 6.2. Each HCICNPCB macro defines a single TCP/IP port through which the HCI is to communicate with partner nodes.

HCICNJCB Macro

There can be zero or more HCICNJCB macros. Each HCICNJCB macro defines a single pre-defined VSAM journal dataset.

HCICNSIT Macro

There can be zero or more HCICNSIT macros. Each HCICNSIT macro defines a single Side Information Table entry. An HCICNSIT macro is used for each conversation allocated outbound from a TP connected to this HCI when that TP is acting as a “client” in a

conversation request. HCICNSIT macros have no bearing on conversations allocated inbound from a remote partner.

HCICNTPT Macro

There can be zero or more HCICNTPT macros. Each HCICNTPT macro defines a single TP Profile Table entry. An HCICNTPT macro is used for each TP that is connected to this HCI when that TP is acting as a "server" in a conversation request. HCICNTPT macros have no bearing on conversations allocated outbound to a remote partner.

The HCICNGCA Macro

Figure 7-4 shows the HCICNGCA macro assembly. When a value is shown to the right of the equal sign, it represents the default value for that operand.

Note: There can be one, and only one, HCICNGCA macro, and it must be the first configuration macro specified.

Figure 7-4. HCICNGCA MACRO: Macro Operands for HCICNGCA.

```
       HCICNGCA SYSID=,
             DAE=YES,
             DFLTUSR=HCIUSER,
             DMPCLAS=,
             DMPPFX=,
             DMPUNIT=,
             DMPVOL=,
             INTERNAL=NO,
             JESCHAR=$,
             JESID=0,
             JESJPRM=NO,
             JESMAIN=,
             JRNMASK=FFFFFFFFFFFFFFFF,
             MAININT=1500,
             MAXCCBS=32,
             MAXDCBS=32,
             MAXJRES=2048,
             MAXLUBS=64,
             MAXRCBS=4,
             MAXUIBS=16,
             MAXWRES=1024,
             OACBINT=3000,
             OPCMD=WTOR,
             PACING=0,
             PREPROC=HCIYPREP,
             ROUTCMD=NO,
             SECFAC=,
             SRVRJOB=HCIJOB,
             SYSPLEX=,
             TCPAMSG=NO,
             TPSEC=NO
```

The HCICNGCA Macro Operands

SYSID=

Specify the four-character SUBSYSTEM_ID that is to be used by this HCI. This operand is required and there is no default. Check with your MVS systems programmer when choosing the SUBSYSTEM_ID, because this value **must not** duplicate any existing SUBSYSTEM_ID. Note that the HCI dynamically creates this subsystem, and does not rely

on and does not allow this name to be included in the IEFSSNxx member of SYS1.PARMLIB. If the HCI finds that this name has been specified in the IEFSSNxx PARMLIB member, the HCI terminates with a User Abend code of U0100 Reason code of 125.

In a SYSPLEX implementation, this operand retains the meaning it had for non-SYSPLEX. The SUBSYSTEM_ID specified for this operand becomes the identification of the HCI on the local MVS system only, and can be used by TPs when those TPs are executing on the same MVS as the HCI is running. As with non-SYSPLEX implementations, this operand value must be unique, but not only within the MVS system, but unique within the complex of MVS systems.

- Default: None. Operand is required.

DAE=

Occasionally it is desirable to bypass the MVS DAE processing and to be able to collect multiple SDUMPs even though the abends represented by these dumps are all identical. The DAE operand specifies that DAE processing should be allowed to process dumps normally (DAE=YES), or to bypass DAE processing (DAE=NO). You should check with your MVS systems programmer before specifying DAE=NO as numerous SDUMPS could affect total system performance.

- Default: YES

DFLTUSR=

Specify a security system USERID to be used for any TP that has no USERID assigned from another source. For LU 6.2 TPs, the USERID can be included in the FMH-5 that flows from the remote partner to the HCI. For LU 2 TPs, the USERID can be included in the Attach Header that flows from the remote partner to the HCI. For TCP/IP, the USERID can be included in the GCS Version Exchange message and/or the UNIFACE Logon message. If at the time the TP is initiated, no USERID has been assigned from one of the above sources, the HCI assigns the value specified here as the default USERID.

This operand is optional. If omitted, DFLTUSR=HCIUSER is used. If this operand is specified as DFLTUSR=, then a default USERID of 8 blanks will be used. If a USERID is specified, it can be from one to seven characters long and must follow the rules of coding dictated by the security subsystem.

The default USERID specified here must be defined to the security subsystem and must have enough authority to initiate the job that is submitted (or started) by the HCI.

Note: For TCP/IP, when neither GCS nor UNIFACE is used, this default user must be a valid USERID.

- Default: HCIUSER

DMPCLAS=

Specify a one-character SYSOUT class for region dumps taken by the HCI. This operand is optional. If omitted, then region dumps will be written to automatically allocated datasets whose characteristics are defined by the macro operands DMPVOL=, DMPUNIT= and DMPPFX=. If this operand is specified then the DMPVOL and DMPPFX operands must be omitted.

- Default: None

DMPPFX=

Specify a 1- to 11-character name that will be used as the prefix for all region dump datasets allocated by the HCI. This operand is optional. If omitted, then DMPCLAS= must be specified and DMPVOL must also be omitted. If this operand is specified, then it must

conform to the rules of coding dataset names. The HCI will add a ' . ' (period) to the end of the characters coded in this operand, and will construct the remainder of the dataset name from there.

Note that the HCI must have authority to create (ACCESS=ALTER) these dump datasets, so the choice of high level qualifiers is important.

If DMPPFX=SDUMP is specified, then the HCI will take dumps to a SYS1.DUMPxx dataset, rather than to the dataset normally allocated. These SDUMPs will be more complete than allocated dataset dumps and can be of more help to Compuware when diagnosing HCI problems.

It is strongly recommended that DMPPFX=SDUMP be coded.

- Default: None

DMPUNIT=

Specify a 1- to 8-character MVS Unit name that the HCI will use when allocating a dataset to contain all region dumps. This operand must be a valid MVS unit name for a DASD device. This operand is required even if DMPPFX=SDUMP is specified, but in this case, it is not used to allocate a SYS1.DUMPxx dataset.

- Default: None. This operand is required.

DMPVOL=

Specify a 1- to 6-character Volume Serial Number of the DASD device that is to contain all region dump datasets. This operand is optional. If omitted, the HCI will make a non-specific volume request when allocating the dataset. If this operand is specified, it must name a DASD volume that is included in the list of volumes owned by the DMPUNIT operand. This operand must be omitted when DMPPFX=SDUMP is specified.

- Default: The system chooses a volume based upon unit name

INTERNAL=

Specify INTERNAL=YES if internal conversations are to be supported without the use of VTAM, or specify INTERNAL=NO if internal conversations are to be supported by using VTAM. An internal conversation is defined as one in which **both** session partners are TPs on the same HCI. This operand is optional. If omitted, the default is INTERNAL=NO. This operand has no meaning for conversations that include one of the conversation partners on this HCI and the other on another HCI or on another network platform.

- Default: NO

JESCHAR=

Specify the control character used by your installation's JES/2 or JES/3 for console commands. This operand is optional. If omitted, JESCHAR=$ is used.

- Default: $

JESID=

Specify JESID=2 if JES/2 is the primary subsystem or specify JESID=3 if JES/3 is the primary subsystem or specify JESID=0 if the HCI is to determine which JES is the primary subsystem. Note that prior to MVS/ESA it was not possible to determine the level of the primary subsystem automatically. Therefore if the HCI is to run on an MVS/XA system, specify this value as JESID=2 or JESID=3.

- Default: 0

JESJPRM=

Specify JESJPRM=YES if the HCI is to supply a control statement immediately following the JOB card in submitted JCL that directs JES/2 or JES/3 to execute the submitted job on the same system as the HCI is running.

For JES/2 systems, this control statement is "/*JOBPARM SYSAFF=*", and for JES/3 systems, this control statement is "//MAIN SYSTEM=*name* ", where *name* is supplied by the JESMAIN= operand in this macro.

Specify JESJPRM=NO if the HCI is not to supply this control statement.

This operand should be omitted or specified as JESJPRM=NO when the HCI is running in SYSPLEX mode, and when you want the TPs submitted by the HCI to run on SYSPLEX members other than the one on which the HCI is running.

- Default: NO

JESMAIN=

Specify the 1- to 8-character name of this JES/3 system. This operand applies **only** if JESJPRM=YES has been specified, and if the HCI is to run on a JES/3 system. This operand supplies the name that the HCI places in the //MAIN SYSTEM=*name* statement that it inserts just after the jobcard in submitted jobs. This operand should be omitted if JESJPRM=NO is specified, or if this is a JES/2 system.

This operand should be omitted if the HCI is running in SYSPLEX mode and you want TPs submitted by the HCI to run on SYSPLEX members other than the one on which the HCI is running.

- Default: None

JRNMASK=

The HCI contains a considerable journaling facility, and this operand controls which HCI events are to be written to the HCI's journal. Specify exactly 16 characters, each of which is a valid HEX digit, 0-9 or A-F. Each hex digit contains 4 bits. All bits have a position starting at the leftmost bit (bit 0), to the rightmost bit (bit 63). If JRNMASK=0000000000000000 is specified, then **NO** journal records will be written. If JRNMASK=FFFFFFFFFFFFFFFF is specified, then **ALL** events will be written to the journal. For your production environment, you should specify all "0's" for this operand since journaling consumes considerable CPU resources. In your test environment, or if requested by a Compuware PSR, you may want to specify all "F's" in order to obtain documentation for a given problem. Do not allow the specification of all "F's" to remain selected longer than is necessary to recreate a problem.

- Default: FFFFFFFFFFFFFFFF

MAININT=

Specify a number in the range of 1 to 360000. This number represents the time in hundredths of seconds between HCI scans for session cleanup processing. The 15 second default for this value is acceptable for all but the most active HCI installations.

- Default: 1500 (15 seconds)
- Minimum value: 1 (1/100 second)
- Maximum value: 360000 (1 hour)

MAXCCBS=

Specify a number in the range of 1 to 4096. This number represents the maximum number of concurrent conversations that can exist at any one time. If a remote workstation attempts to establish a conversation or connection with the HCI, and the

HCI is currently at maximum number of conversations, the conversation request will be denied.

- Default: 32
- Minimum value: 1
- Maximum value: 4096

MAXDCBS=

Specify a number in the range of 1 to 1024. This number represents the maximum number of different remote destinations with which the HCI can be communicating at any one time. A remote LU 6.2 workstation represents a single destination, even though it may have multiple LU 6.2 conversations active with the same HCI. A remote TCP/IP workstation represents a single destination, even though it may have multiple TCP/IP connections active with the same HCI. If a remote workstation attempts to establish a conversation or connection with the HCI, and the HCI is currently at its maximum number of destinations, the conversation request will be denied. The request can be retried at a later time and will be accepted if the HCI is no longer at its maximum destination limit.

- Default: 32
- Minimum value: 1
- Maximum value: 1024

MAXJRES=

Specify a number in the range of 1 to 8191. This number represents the maximum number of journal records that can be queued within main storage (waiting to be written to the journal) at any one time. If you have specified the journal mask as all zeros, you only need to specify MAXJRES=1; but if you have specified a non-zero journal mask, then this number should be close to the maximum, which is 8191. If a module within the HCI attempts to write a journal record, and the HCI is currently at the maximum number of queued records, the attempt will fail, and the journal record will be lost.

- Default: 2048
- Minimum value: 1
- Maximum value: 8191

MAXLUBS=

Specify a number in the range of 1 to 1024. This number represents the maximum number of concurrent LU 2 sessions that can be active with the HCI at any one time. Each remove LU 2 workstation represents one LU 2 session, and requires one LUB. If an LU 2 workstation attempts to establish a session with the HCI, but the HCI is already servicing the maximum number of workstations, the new session request will be denied.

- Default: 64
- Minimum value: 1
- Maximum value: 1024

MAXRCBS=

Specify a number in the range of 1 to 256. This number represents the maximum number of concurrent address spaces that can be active with the HCI at any one time. Each address space can contain one or more HCI TPs. If any remote workstation attempts to initiate a new conversation with the HCI, and that conversation requires a new address space, and the HCI is already servicing the maximum number of address spaces, the new request will be denied. The request can be retried at a later time and will be accepted if the HCI is no longer at its maximum address space limit.

- Default: 4
- Minimum value: 1
- Maximum value: 256

MAXUIBS=

Specify a number in the range of 1 to 1024. This number represents the maximum number of concurrent users (or TPs) across all address spaces (HCI) at any one time. Each address space can contain one or more TPs that can be active at any one time. If any remote workstation attempts to initiate a new conversation with the HCI, and the HCI is already servicing the maximum number of users, the new request will be denied.

- Default: 16
- Minimum value: 1
- Maximum value: 1024

MAXWRES=

Specify a number in the range of 1 to 4096. This number represents the maximum number of concurrent work requests that can be active at any one time within the HCI. This is a critical value since the HCI will abend if it attempts to obtain a WRE and one is not available. Start with a value close to the maximum, and decrease this value, over time, as experience shows that the HCI is executing successfully. The WREs are allocated above the line and each has a size of 256 bytes.

- Default: 1024
- Minimum value: 1
- Maximum value: 4096

OACBINT=

Specify a number in the range of 1 to 360000. This number represents the time in hundredths of a second, that the HCI will wait between unsuccessful attempts to open a VTAM ACB or to initiate TCP/IP communications. This value is used when a VTAM ACB OPEN fails. When the time represented by this value expires, the HCI will try to open the ACB or initiate the TCP/IP connection again. The value specified for OACBINT must be greater than the value specified for MAININT and must be a multiple of that value.

- Default: 3000 (30 seconds)
- Minimum value: 1 (1/100th second)
- Maximum value: 360000 (1 hour)

OPCMD=

Specify the method by which commands are to be entered to the HCI. Specify OPCMD=WTOR, if a Write to Operator with Reply (WTOR) is to be left outstanding on the MVS master console. Specify OPCMD=MODIFY if the MVS MODIFY command is to be supported. Or specify OPCMD=NONE if no operator commands are to be entered from the master console.

Note: For HCI versions 1.1 through 1.7, only OPCMD=WTOR or OPCMD=NONE are valid. For HCI version 2.0 and later, any of the three options may be specified.

- Default: WTOR

PACING=

Specify a number in the range of 0 to 2048. This number represents the number of messages that can flow between internal TP tasks before a pacing response is required from the receiving task.

Note: This operand is not used for HCI versions 1.1 through 2.2.

- Default: 0
- Minimum value: 0
- Maximum value: 2048

PREPROC=

Specify the 1- to 8-character name of the procedure you have stored in SYS1.PROCLIB that invokes the SYSPLEX initialization program, HCIYPREP. (Refer to the *HCI User/Reference Guide* for a complete description of the functions that this program performs). This operand exists so that you can have different versions of the HCI installed on the same MVS at the same time. In order to execute the HCIYPREP program of the appropriate version, different PROCs are required, and this operand allows you to specify which PROC represents the correct version. If this operand is omitted, PREPROC=HCIYPREP is used as a default.

- Default: HCIYPREP

ROUTCMD=

Specify ROUTCMD=YES if you want the HCI to start TPs not only on the local MVS system, but on the other MVS systems in the SYSPLEX as well. Specify ROUTCMD=NO if you want the HCI only to start TPs on the local MVS system. (Refer to the *HCI User/Reference Guide* for a complete description of the HCI's ability to start TPs on multiple MVS systems).

- Default: NO

SECFAC=

Specify a 1- to 64-character facility-class entity name, which can be used to secure the HCI console commands that can be issued by TPs. If this parameter is coded, the HCI will make a RACROUTE call whenever a TP issues a console command to ensure that the USERID associated with the TP has access to the named facility-class entity. READ access is required to issue the DISPLAY and SDUMP console commands. UPDATE access is necessary for all other console commands.

Note: This parameter has no effect on HCI console commands that are entered by the console operator.

For information on HCI console commands, refer to the *HCI User/Reference Guide*.

- Default: None. If this operand is omitted, no RACROUTE call will be made.

SRVRJOB=

Specify a one- to seven-character job name prefix that the HCI will use as the basis for the JOBNAME on all TP jobs submitted by the HCI. The HCI will supply the remaining characters of the JOBNAME, a numeric suffix, to make an eight-character name with numbers sequentially numbered from 1 to n. The number of characters specified for this name should leave enough room for enough numeric digits to guarantee uniqueness of JOB names. That is, if a seven-character name were supplied for this operand, then only ten concurrent TPs could execute. Or, provide a one- to seven-character job name ending with a “-”. This will cause the numeric suffix to be omitted

Specify SRVRJOB=, (to define the macro operand as omitted) if you want server jobs to be named with the USERID of the submitter at the client side of the conversation. When SRVRJOB is defined in this way, the JOBNAME of all submitted TPs and all started TPs will start with the USERID of the submitter, followed by a sequential number 0, 1, 2, etc.

Note: For TCP/IP communications over a port defined as RECFM=U, and **not** defined as PROTOCOL=GCS, PROTOCOL=UNIFACE nor PROTOCOL=UNIFACE8, SRVRJOB **MUST** be defined with a valid job name prefix.

- Default: HCIJOB

SYSPLEX=

Specify this operand, followed by the four-character SUBSYSTEM_ID that represents the HCI on the SYSPLEX, if you want the HCI to run in SYSPLEX mode. Specify SYSPLEX=, (or omit this operand) if you want the HCI to run in non-SYSPLEX mode. The SUBSYSTEM_ID specified here must be different from the SUBSYSTEM_ID specified in the SYSID= operand and must be unique across all MVS systems in the SYSPLEX. Like the SYSID=SUBSYSTEM_ID, this subsystem cannot have been included in the IEFSSNxx member of SYS1.PARMLIB.

- Default: SYSPLEX= (no value means no SYSPLEX support)

TCPAMSG=

Specify either TCPAMSG=YES or TCPAMSG=NO, or omit this operand. TCPAMSG applies only to installations that are running the INTERLINK TCP/IP product. TCPAMSG=YES specifies that TCP/IP related messages that are displayed on the MVS master console are to be displayed in the INTERLINK TCPaccess format. TCPAMSG=NO specifies that TCP/IP related messages that are displayed on the MVS master console are to be displayed in the HCI standard format. You should specify TCPAMSG=YES if your operations staff is used to seeing the TCPaccess messages and is used to interpreting the values specified in these messages.

- Default: NO

TPSEC=

Specify either TPSEC=YES or TPSEC=NO, or omit this operand. TPSEC=YES instructs the HCI to validate, via RACROUTE, each TPs registration request to ensure that the USERID/PASSWORD under which the TP is running has authority to register as the TP_NAME specified in the register request. A FACILITY CLASS entity must be established, which is the concatenation of the HCI SUBSYSTEM_ID with the TP_NAME to be checked. Separate the SUBSYSTEM_ID from the TP_NAME with a period ('.'). Each USERID that can run that TP must have READ access to the FACILITY CLASS entity. If the USERID does not have READ access to this entity, the registration request is denied. If you specify TPSEC=NO, then no checking is done to ensure that a TP has authority to run under the TP_NAME it is registering as.

- Default: NO

The HCICNAMB Macro

Figure 7-5 shows the HCICNAMB macro assembly.

Note: There can be zero or more HCICNAMB macros. For LU 2 and LU 6.2, each HCICNAMB macro defines a unique VTAM ACB, which must have a corresponding APPL statement in the VTAMLST dataset. For TCP/IP, each HCICNAMB defines a connection with a single TCP/IP address space. If an installation runs only one TCP/IP address space, then there would be only one HCICNAMB macro corresponding to the single TCP/IP. If an installation runs more than one TCP/IP address space, and if the HCI is to communicate with more than one TCP/IP at a time, then a separate HCICNAMB macro must be coded for each TCP/IP address space.

Figure 7-5. HCICNAMB MACRO: Macro Operands for HCICNAMB.

```
       HCICNAMB ,
             ACBNAME=,
             GENNAME=,
             LUTYPE=,
             HPNS=NO,
             MODEL=0,
             RECVLEN=4096,
             TCPNAME=,
             TCPPSWD=
```

The HCICNAMB Macro Operands

ACBNAME=

Specify a 1- to 8-character name that is to be used by the HCI as the name for an ACB, which is to be opened and used for communications from remote partners. This operand is required for LUTYPE=LU62 and LUTYPE=LU2 and is not used for LUTYPE=TCPIP. This name must match the ACBNAME= operand on an APPL statement in the SYS1.VTAMLST concatenation.

Note: If, on the APPL statement, the name of the APPL definition is different from the ACBNAME= operand, then the remote partner LU must establish a session using the **name on the APPL statement** not the ACBNAME= value.

- Default: None. Required for LU62 and LU2. Not used for TCPIP.

GENNAME=

Specify a 1- to 8-character name to invoke VTAM's generic name support for the ACB defined in this macro. If you omit this operand, no generic name support is supplied. This operand applies only to LU 2 and LU 6.2 ACBs, and is ignored for TCP/IP connections. VTAM's generic name support allows multiple VTAM APPLs to be known externally by a single name, and VTAM will assign each incoming session request to a particular ACB based upon the number of sessions already established with the same generic name. You must have an active SYSPLEX COUPLING facility installed on your MVS systems for this operand to have any effect. You may assign the same generic name to multiple ACBs within a single HCI, and/or you may assign the same generic name to multiple ACBs across different HCIs. A single generic name must contain all LU 2 or all LU 6.2 ACBs: You cannot mix LU 2 with LU 6.2. When this support is activated, remote LUs request sessions with the GENERIC NAME, rather than the APPLID name.

- Default: None

LUTYPE=

Specify one of the following values that defines the type of communications that this HCICNAMB macro represents. Code LUTYPE=LU2 for GCS communications with an HLLAPI emulator, which uses the LU 2 protocol. Code LUTYPE=LU62 for communications using LU 6.2 (a.k.a. APPC) protocol. Code LUTYPE=TCPIP for communications using the IBM TCP/IP for MVS product. Code LUTYPE=TCPACCESS for communications using the Interlink Corporation's TCP/IP product.

- Default: None. This operand is required.

HPNS=

Specify HPNS=YES, if the High Performance Network Socket feature of TCP/IP for MVS Version 3 Release 2 should be enabled, or specify HPNS=NO if not. Note that HPNS=YES can ONLY be specified if Version 3 Release 2 of TCP/IP for MVS is installed. If the

appropriate PTF for APAR PN90450 is installed, then you can specify either HPNS=YES or HPNS=NO for V3R2 of TCP/IP. If this PTF is not installed on V3R2 of TCP/IP then you must specify HPNS=YES. If you are running V3R1 of TCP/IP, then you must specify HPNS=NO, or allow it to default.

- Default: NO

Table 7-1. Specification of HPNS=: Alternatives Affecting HPNS

PTF	IBM V3R1	IBM V3R2	INTERLINK
UN93760 not installed	Can't run	Not applicable	HCI 2.1 and above
UN93760 installed	HCI 1.8 and above with HPNS=NO	Not applicable	HCI 2.1 and above
PTF for PN90450 not installed	Not applicable	HCI 1.9 and above with HPNS=YES	HCI 2.1 and above
PTF for PN90450 installed	Not applicable	HCI 1.8 and above with HPNS=YES or HPNS=NO	HCI 2.1 and above

MODEL=

Specify a numeric value, 0, 2, 3, 4 or 5 or omit this operand. If this operand is omitted, or if MODEL=0 is specified, the HCI will dynamically determine the LU 2 model type at the time that the LU establishes a session with the HCI. Some 3270 emulators, however, do not support the method the HCI uses to make this determination. You can cause the HCI to bind the LU 2 as a particular model by specifying the model number in this operand.

- Default: 0

RECVLEN=

Specify a numeric value (in the range of 512 to 16777211) that represents the default size of the maximum data buffer that can be received on the protocol defined by this HCICNAMB macro. Note that this is a default value, and can be overridden, either upward or downward, by the RECVLEN operand on an HCICNTPT macro. This value is used only when no overriding value from an HCICNTPT macro exists.

- Default: 4096
- Minimum value: 512
- Maximum value: 16,777,211

TCPNAME=

Specify this operand **only** when this HCICNAMB macro is defining the TCP/IP protocol and specifies LUTYPE=TCPIP or LUTYPE=TCPACCESS.

If LUTYPE=TCPIP, then this is a 1- to 8-character name that must match the **TCPIPJOBNAME operand in the TCPIP.DATA member that is specified as DDNAME SYSTCPD in the procedure that is used to start the TCP/IP address space**.

If LUTYPE=TCPACCESS, then this is a 4-character name that must match the **UNIQ= operand in the PARM field of the EXEC statement in the procedure that invokes Interlink's TCPaccess product**.

This operand is required. If this name is specified incorrectly, the HCI will abend during initialization with a User Abend code U200 and a reason code of X'000000D0' (208). This abend indicates that there is no TCP/IP address space active with this TCPNAME but that there is **some** TCP/IP address space active. Check carefully to ensure that this operand is specified correctly. The HCI can be started prior to starting TCP/IP, and will try repeatedly to establish communications with the TCP/IP region. When TCP/IP is started, if this operand is specified correctly, a connection will be made and communications can commence.

- Default: None. This operand is required.

TCPPSWD=

Code this operand only if:

1. LUTYPE=TCPACCESS for this HCICNAMB macro **and**
2. The TCPaccess password facility has been invoked in the TCPaccess customization dataset.

- Default: No password supplied

The HCICNPCB Macro

Figure 7-6 shows the HCICNPCB macro assembly.

Note: There can be zero or more HCICNPCB macros. This macro applies to TCP/IP only, and is not used for LU 2 or LU 6.2.

Figure 7-6. HCICNPCB MACRO: Macro Operands for HCICNPCB.

```
       HCICNPCB ,
             DATAENC=ASCII,
             EOBCHRS=,
             EORCHRS=,
             LLZZ=,
             LRECL=4096,
             PORT=,
             PROTOCOL=,
             RECFM=,
             TCPNAME=,
             TPNAME=,
             TYPE=LOCAL
```

Each HCICNPCB macro defines a single TCP/IP PORT that can have one of two uses: If TYPE=LOCAL is specified (or allowed to default), then the HCI will “listen” on this port for incoming connection requests, and will establish TPs to process the data flowing over this port. This specification defines the TP to be the “server” end of a TCP/IP connection. If TYPE=REMOTE is specified, the HCICNPCB macro represents the remote end of a TCP/IP connection established by the HCI. All operands on this HCICNPCB macro refer to the **partner** node, and not to the **local** node. An HCICNPCB macro with TYPE=REMOTE specified must be named in an HCICNSIT entry. HCICNPCB macros with TYPE=REMOTE are only used for connections established by the HCI, where the TP in question is to act as the “client” end of a TCP/IP communication.

The HCICNPCB Macro Operands

DATAENC=

Specify either ASCII or EBCDIC to define the data representation used for communications through the port defined by this HCICNPCB macro. If PROTOCOL=GCS, PROTOCOL=UNIFACE or PROTOCOL=UNIFACE8 is specified, then this operand should be omitted since the PROTOCOL specification overrides any other macro operands.

- Default: ASCII
- Not used if PROTOCOL=GCS, PROTOCOL=UNIFACE or PROTOCOL=UNIFACE8

EOBCHRS=

Specify the 1- to 16-byte character string that is used to define the end of each block of messages received by the HCI from the remote partner. Specify this operand **only** if:

- RECFM=U is specified

and none of the following are specified:

- PROTOCOL=GCS
- PROTOCOL=UNIFACE
- PROTOCOL=UNIFACE8

These characters are specified as two-byte hexadecimal digits and must be specified in the encoding scheme that is used for the communications. Each message terminated by these EOBCHRS will be considered a complete message, and will be passed to the appropriate TP as “complete data”.

If RECFM=U is specified, and if EOBCHRS is NOT specified, then the HCI can make no assumptions about the completeness or incompleteness of any inbound message, and must pass whatever amount of data it has to the TP with the indication of “data received”, which indicates to the TP that it must process the data regardless.

- Default: None

EORCHRS=

Specify the 1- to 16-byte character string that is used to define the end of each record received by the HCI from the remote partner. Specify this operand **only** if:

- RECFM=U is specified

and none of the following are specified:

- PROTOCOL=GCS
- PROTOCOL=UNIFACE
- PROTOCOL=UNIFACE8

These characters are specified as two-byte hexadecimal digits and must be specified in the encoding scheme that is used for the communications. Each record terminated by these EOBCHRS will be considered an incomplete message, and will be passed to the appropriate TP as “incomplete data”.

If RECFM=U is specified, and if EORCHRS is **NOT** specified, then the HCI can make no assumptions about the completeness or incompleteness of any inbound message, and it must pass whatever amount of data it has to the TP with the indication of “data received”, which indicates to the TP that it must process the data regardless.

- Default: None

LLZZ=

Specify a set of macro operands, enclosed in parentheses, that define the location and format of the length indicator area within each TCP/IP message that flows into or out of the HCI on this port. If PROTOCOL=GCS, PROTOCOL=UNIFACE or PROTOCOL=UNIFACE8 is specified, then this operand should be omitted since the protocol specification overrides any other macro operands.

The sub-operands within this operand specify the following:

```
LLZZ=(start_loc,length,data_rep,scope,out_rule,in_rule)
```

start_loc: A numeric value that specifies the starting location in each TCP/IP message of the length indicator field. This value is relative to 0. Therefore, it can have a value of 0

(which specifies the first byte of each message) up to the maximum length of the message.

length: A numeric value that specifies the length in bytes of the LLZZ field. The minimum length is 1 byte, and the maximum length is 8 bytes. Remember, this is the length of the LLZZ field within each data message, and not the length of the message itself.

data_rep: A literal that can have one of four values: “CHARA”, “CHARE”, “BINB” or “BINL”.

- CHARA specifies that the LLZZ field contains ASCII encoded numbers.
- CHARE specifies that the LLZZ field contains EBCDIC encoded numbers.
- BINB specifies that the LLZZ field contains binary numbers in the “Big Endian” (or network byte order) format.
- BINL specifies that the LLZZ field contains binary numbers in the “Little Endian” format.

scope: A literal that can have one of two values: “INCL” or “EXCL”.

- INCL specifies that the numeric value contained in the LLZZ field **includes** itself (in the way that the LLZZ field on variable length records includes itself).
- EXCL specifies that the numeric value contained in the LLZZ field **excludes** itself (in that it defines the number of bytes **following** the LLZZ field itself).

out_rule: A literal that can have one of two values: “MAKE” or “USE”.

- MAKE specifies that the HCI is to create the LLZZ field in each output message.
- USE specifies that the TP has created the LLZZ field, and the HCI is to use this already-created LLZZ.

in_rule: A literal that can have one of two values: “DELETE” or “KEEP”.

- DELETE specifies that the HCI is to remove the LLZZ field from all inbound messages before presenting the message to the TP.
- KEEP specifies that the HCI is to pass the LLZZ field, intact, to the TP.

- Default: None
- Not used if PROTOCOL=GCS, PROTOCOL=UNIFACE or PROTOCOL=UNIFACE8 is specified.

LRECL=

Specify a numeric value (in the range of 512 to 32760) that represents the largest logical message that can be received from or sent to the port represented by this HCICNPCB macro. This value does not have to match the RECVLEN= operand on the HCICNAMB or the HCICNTPT macros. It simply places a limit on message size for the defined port. If PROTOCOL=GCS, PROTOCOL=UNIFACE or PROTOCOL=UNIFACE8 is specified, then this operand should be omitted since the protocol specification overrides any other macro operands.

- Default: 4096
- Minimum Value: 512
- Maximum Value: 32760
- Not used if PROTOCOL=GCS, PROTOCOL=UNIFACE or PROTOCOL=UNIFACE8 is specified.

PORT=

Specify the number that defines the TCP/IP PORT represented by this HCICNPCB macro. This number is required in all cases, regardless of the PROTOCOL= specification. This number can range from 0 to nnnnn (TBD).

- Default: None. This operand is required.

PROTOCOL=

This operand specifies the protocol to be used for all communication through the PORT defined by this HCICNPCB macro. Specify one of the following to process the message traffic:

- GCS (for all GCS versions)
- UNIFACE (if the version is prior to version 8)
- UNIFACE8 (if the version is 8 or later)

Omit this operand for any other message processing software.

- Default: None.

RECFM=

This operand defines the overall record format of each message processed through the PORT defined by the HCICNPCB macro. Specify one of the following:

- Specify RECFM=F if each message (as defined by the LRECL= operand) is of a fixed length.
- Specify RECFM=V if messages are of a variable length, and if each contains a length field (as defined by the LLZZ operand).
- Specify RECFM=U if messages are of an undefined length and are terminated by end-of-record and end-of-block character strings.

If PROTOCOL=GCS, PROTOCOL=UNIFACE or PROTOCOL=UNIFACE8, is specified, then this operand should be omitted since the protocol specification overrides any other macro operands.

- Default: None.

TCPNAME=

Specify the name of the TCP/IP address space that is to process the messages through the PORT defined by the HCICNPCB macro. This operand is only required if there is more than one HCICNAMB macro that specifies LUTYPE=TCPIP, and the name specified here must match the TCPNAME= operand on the associated HCICNAMB macro.

- Default: None. Not required if only one HCICNAMB macro with LUTYPE=TCPIP exists.

TPNAME=

This operand specifies the symbolic_destination_name (SYMDEST) of the TP Profile table (HCICNTPT) entry that is to be used for all connections using the PORT number defined by this HCICNPCB macro. This operand is **only** used when PROTOCOL=UNIFACE or PROTOCOL=UNIFACE8 is specified, or when the PROTOCOL= operand is omitted. **Omit this operand when PROTOCOL=GCS is specified.** This name must match the SYMDEST= name on an HCICNTPT macro entry. Whenever connection requests arrive on this port, the TP defined by the HCICNTPT macro will be used to process the message traffic.

- Default: None. Not required if PROTOCOL=GCS or PROTOCOL= is omitted

TYPE=

This operand specifies whether the macro operands on this HCICNPCB macro refer to LOCAL or REMOTE processing. LOCAL processing implies that the PORT defined in this macro is a “server” port, and the HCI will “listen” on this port for incoming connection requests. REMOTE processing implies that the PORT defined in this macro is a “server” port **ON A REMOTE NODE**. The HCI will not “listen” for connection requests on ports defined as REMOTE.

- Default: LOCAL

The HCICNJCB Macro

Figure 7-7 shows the HCICNJCB macro assembly.

Note: There can be zero or more HCICNJCB macros.

Figure 7-7. HCICNJCB MACRO: Macro Operands for HCICNJCB.

```
       HCICNJCB ,
             DDNAME=,
             DSNAME=
```

Each HCICNJCB macro defines a single HCI journal dataset. Even if you have specified JRNMASK=0000000000000000, which keeps the HCI from writing any journal records, you should specify at least one journal dataset, against the time that you want to run the journal facility.

The HCICNJCB Macro Operands

DDNAME=

Specify a 1- to 8-character name that the HCI will use as the DDNAME for this journal dataset. This name is used in console messages to relate the message to a particular journal dataset. There are no requirements for this name to match any other name. It must simply be unique within this HCI.

- Default: None

DSNAME=

Specify the 1- to 44-character DSNAME for this journal dataset. This name must have been used in the IDCAMS DEFINE CLUSTER job used to define the VSAM journal dataset.

- Default: None

The HCICNSIT Macro

Figure 7-8 shows the HCICNSIT macro assembly.

Note: There can be zero or as many HCICNSIT macros as desired.

Figure 7-8. HCICNSIT MACRO: Macro Operands for HCICNSIT.

```
        HCICNSIT ,
              SYMNAME=,
              BYPDLAY=NO,
              IPADDR=,
              IPNAME=,
              LLUNAME=,
              LUTYPE=LU62,
              MAXS=0,
              MINCL=0,
              MINCW=0,
              MODNAME=,
              PARSESS=YES,
              PASSWD=
              PLUNAME=,
              PORT=,
              TCPNAME=,
              TPNAME=,
              USERID=
```

The set of all HCICNSIT macros in the configuration assembly comprise the HCI SIDE INFORMATION TABLE. Each HCICNSIT macro defines a single Side Information Table Entry, and each defines a single connection made by a TP running on this HCI with a remote partner. An HCICNSIT entry is required for each connection made **outbound** from this HCI to a remote partner. HCICNSIT entries play no part in connections that are established by the remote partner to a TP on this HCI.

The HCICNSIT Macro Operands

SYMNAME=

Specify the symbolic destination name (SYMDEST_NAME) that is to become the key for this HCICNSIT entry. It is this name that the TP uses in its CMINIT CPI-C call. Therefore, this name must be known to the TP that is going to establish a connection/conversation with the partner node.

- Default: None. This operand is required

BYPDLAY=

Specify whether the HCI is to wait after issuing a TCP/IP SHUTDOWN before issuing socket CLOSE. Specify BYPDLAY=YES for Uniface applications. Specify (or default to) BYPDLAY=NO for all other TCP/IP applications. This operand is valid **only** for TCP/IP communication.

- Default: NO

IPADDR=

Specify the IP address, in dotted decimal notation, of the remote partner with whom the connection represented by this SYMNAME is to take place. This operand is valid only for TCP/IP communication, and if IPADDR= is specified, then IPNAME must be omitted. Dotted decimal notation takes the form of nnn.nnn.nnn.nnn, where nnn is a 1 to 3 digit decimal number in the range of 0 to 255.

- Default: None

IPNAME=

Specify the IP name that is known to TCP/IP and/or to the local domain name server that represents the remote partner with whom the connection request represented by this SYMNAME is to take place. This operand is valid **only** for TCP/IP communication, and if IPNAME= is specified, then IPADDR must be omitted.

- Default: None

LLUNAME=

Specify the VTAM ACBNAME that is to be used as the local LU name for the conversation represented by this SYMNAME. This operand is valid **only** for LU 6.2 communications. The name specified must match an ACBNAME= operand on an HCICNAMB entry in this HCI configuration assembly. Remember that this name is for the local (this HCI) application name and becomes the VTAM LU name for the local end of the LU 6.2 conversation.

- Default: None

LUTYPE=

Specify:

- LUTYPE=LU62 for LU 6.2 conversations or LUTYPE=LU2 for LU 2 sessions
- Specify LUTYPE=TCPIP for conversations using the IBM TCP/IP product
- Specify LUTYPE=TCPACCESS for conversations using the Interlink TCPaccess TCP/IP product.
- Default: LU62

MAXS=

Specify the maximum number of sessions that can exist between the HCI and remote partner for the mode name specified in this macro. This operand is required only for LU 6.2 communications and must be omitted for LU 2 and TCP/IP connections. The minimum value for MAXS 2 is 0, and the maximum value is 99999.

- Default: 0
- Minimum Value: 0
- Maximum Value: 99999

MINCL=

Specify the minimum number of contention loser sessions that are to exist between this HCI and the remote partner. This operand is required only for LU 6.2 communications and must be omitted for LU 2 and TCP/IP connections. The minimum value for MINCL is 0, and the maximum value is 255.

- Default: 0
- Minimum Value: 0
- Maximum Value: 255

Note: The configuration operands, **MAXS, MINCL and MINCW should always be specified as 0 (zero)** unless VTAM is a release before release 3.2. The specification of zero for these three operands will indicate to the HCI to use the CNOS values specified via the APPL definition statement in VTAMLST.

MINCW=

Specify the minimum number of contention winner sessions that are to exist between this HCI and the remote partner. This operand is required only for LU 6.2 communications, and must be omitted for LU 2 and TCP/IP connections. The minimum value for MINCW 2 is 0 and the maximum value is 255.

- Default: 0
- Minimum Value: 0
- Maximum Value: 255

MODNAME=

Specify the 1- to 8-character logon mode table entry name that is to be used to define the sessions between this HCI and the remote partner LU. This is not the name of the mode table itself, but rather it is the name of the entry within a mode table. The name of the table must be specified on the APPL statement that defines the local LU name (LLUNAME) for this HCI. This operand is valid only for LU 6.2 communications, and should be omitted for LU 2 and TCP/IP connections.

- Default: None

PARSESS=

Specify PARSESS=YES if, for LU 6.2 communications, parallel sessions are to be established between this HCI and the remote partner. This operand is ignored for LU 2 and TCP/IP connections. PARSESS=NO can be specified for LU 6.2 sessions if the remote partner LU does not support parallel sessions.

- Default: YES

PASSWD=

Specify the password that is to be placed into the outgoing FMH-5 for LU 6.2 sessions represented by this SYMNAME entry. This operand provides an alternate method for choosing a USERID/PASSWORD pair for conversations initiated by a TP on this HCI. The preferred method is to allow the TP to supply the password at the time the CMINIT is issued.

- Default: None

PLUNAME=

Specify the Partner_LU_Name that represents the remote partner in the LU 6.2 conversation. This name is used only for LU 6.2 connections and should be omitted for LU 2 and TCP/IP connections. This name need not be pre-defined to the local VTAM if APPN is supported, or if adjacent SSCPs are supported. This name will be defined to VTAM either as an APPL in this or another domain, or as an LU in this or another domain. This name can be specified as NETID.LUNAME or simply LUNAME.

- Default: None

PORT=

Specify the port number in the remote partner node to be used for connections represented by this SYMNAME. This operand is used only for TCP/IP connections, and must be omitted for LU 6.2 and LU 2 communications. The value for port must match the port number in the remote partner on which the LISTEN is executed.

- Default: None

TCPNAME=

Specify this operand **only** when this HCICNSIT macro is defining the TCP/IP protocol and specifies LUTYPE=TCPIP. This is a 1- to 8-character name that must match the TCPIPJOBNAME operand in the TCPIP.DATA member, which is specified as DDNAME SYSTCPD in the procedure that is used to start the TCP/IP address space. This operand is required. If this name is specified incorrectly, the HCI will abend during initialization with a User Abend code U200 and a reason code of X'000000D0' (208). This abend indicates that there is no TCP/IP address space active with this TCPIPJOBNAME, but that there is **some** TCP/IP address space active. Check carefully that this operand is specified

correctly. The HCI can be started prior to starting TCP/IP and will try repeatedly to establish communications with the TCP/IP region. When TCP/IP is started, if this operand is specified correctly, a connection will be made and communications can commence.

- Default: None. This operand is required.

TPNAME=

Specify the 1- to 64-character TP name, as it is known in the remote domain, which is to be started and to which the conversation associated with this HCICNSIT entry is to be associated. Except for Uniface, this operand applies to LU 6.2 and internal conversations only and is ignored for LU 2 and TCP/IP connections.

- Default: None

USERID=

Specify the Userid that is to be placed into the outgoing FMH-5 for LU 6.2 sessions represented by this SYMNAME entry. This operand provides an alternate method for choosing a USERID/PASSWORD pair for conversations initiated by a TP on this HCI. The preferred method is to allow the TP to supply the password at the time the CMINIT is issued.

- Default: None

The HCICNTPT Macro

Figure 7-9 shows the HCICNTPT macro assembly.

Note: There can be zero or more HCICNTPT macros.

Figure 7-9. HCICNTPT MACRO: Macro Operands for HCICNTPT.

```
       HCICNTPT ,
             TPNAME=,
             ALIASOF=,
             CATEGORY=,
             MAXCNV=1,
             MAXPTIM=30,
             MAXSTIM=30,
             MAXTPS=999,
             MEMBER=,
             MULTSEC=NO,
             RECVLEN=,
             SRVRJOB=,
             USERSEC=YES,
             WILDCRD=NO
```

The set of HCICNTPT macros in this configuration assembly comprise the HCI's TP PROFILE TABLE. Each HCICNTPT macro defines a single TP that is to be controlled by this HCI. An HCICNTPT macro is required for each inbound connection request for LU 6.2, LU 2 or TCP/IP. It is via this table that the HCI determines what JCL to submit (or procedure to start), and what the execution characteristics of each TP are.

The HCICNTPT Macro Operands

TPNAME=

Specify the 1- to 64-character TP_NAME for this entry. This name is contained in the incoming FMH-5 for LU 6.2 conversations, in the Attach Header for LU 2 connections, or in the Connections Binding for Uniface or GCS TCP/IP connections. Additionally, this name can be specified in the HCICNPCB macro when a permanent relationship between a TCP/IP port and a particular TP is desired. This operand is required.

- Default: None. This operand is required.

ALIASOF=

Specify the TP_NAME, which must be defined in another HCICNTPT macro for which **this** TP is to become an alias. The remaining operands on this HCICNTPT macro have no effect on the execution of the TP because the values for those operands are taken from the "real" HCICNTPT macro.

- Default: ALIASOF= (operand omitted)

CATEGORY=

Specify a 1- to 64-character category name for this TP. The category name can be used when displaying a list of active TPs.

- Default: CATEGORY= (operand omitted)

MAXCNV=

Specify the maximum number of conversations that each instance of the TP represented by this HCICNTPT macro can support. The minimum value is 1 and the maximum is 999.

- Default: 1
- Minimum Value: 1
- Maximum Value: 999

MAXPTIM=

Specify the time in seconds (in the range of 1 to 1440) that the HCI will wait for a TP to register after the HCI has received the connection request from the remote partner but before the HCI is able to submit (or start) the TP. If this timer expires, the remote (initiating) partner is notified that the connection request has failed, and all storage associated with the new TP is freed.

- Default: 30
- Minimum Value: 1
- Maximum Value: 1440

MAXSTIM=

Specify the time in seconds (in the range of 1 to 1440) that the HCI will wait for a TP to register after the HCI has received the connection request from the remote partner and after the HCI has submitted (or started) the TP. If this timer expires, the remote (initiating) partner is notified that the connection request has failed, and all storage associated with the new TP is freed. Additionally, an MVS CANCEL command is issued to ensure that the TP does not start after this timer has expired.

- Default: 30
- Minimum Value: 1
- Maximum Value: 1440

MAXTPS=

Specify the maximum number of instances of this TP that are allowed to be active on the MVS system at one time. The minimum value is 1 and the maximum value is 999. If the maximum number is reached, subsequent connection requests for this TP are rejected until the number running falls below this maximum value. For example, UMPIRE job can handle 40 polyserver tasks. If a high water limit of 200 client conversations is the objective, then MAXTPS should be set to 5 (jobs).

- Default: 999
- Minimum Value: 1
- Maximum Value: 999

MEMBER=

Specify the PDS member name within the HCI PARMLIB that contains the JCL to be submitted (or the name of the PROC to be started) to invoke each instance of this TP. This operand is required whenever the HCI is to invoke the TP. If the TP always starts by itself, then this operand can be omitted.

- Default: None

MULTSEC=

Specify MULTSEC=YES if the TP represented by this HCICNTPT macro supports multiple simultaneous conversations and if the HCI is to automatically reassign the security environment on behalf of the TP. Specify MULTSEC=NO if the TP represented by this HCICNTPT macro does not support multiple simultaneous conversations or if the HCI is not to provide automatic security context processing.

- Default: NO

RECVLEN=

Specify the size of each input receive buffer allocated by the HCI for this TP. This operand can be omitted, in which case the RECVLEN value from the HCICNAMB macro is used, or this operand can override the HCICNAMB value on a TP by TP basis. The values specified here conform to the same rules as for RECVLEN on the HCICNAMB macro.

- Default: None

SRVRJOB=

Specify the JOBNAME prefix that is to be used for TPs defined by this HCICNTPT entry and submitted by the HCI. The value specified here is used instead of the value specified on the HCICNGCA macro. The JOBNAME is constructed in the same method as that of the HCICNGCA entry. If this operand is omitted, or specified as SRVRJOB=, or allowed to default, then the value specified on the HCICNGCA macro is used.

- Default: SRVRJOB=, (same as operand omitted)

USERSEC=

Specify USERSEC=NO if the associated TP is not to have any RACROUTE processing performed for it by the HCI. Specify USERSEC=YES, or omit the operand, if the HCI is to perform normal security processing. All other TPs should omit this operand.

- Default: YES (same as operand omitted)

WILDCRD=

Specify WILDCRD=YES if this HCICNTPT macro represents a generic name for a set of TPs. That is, the TPNAME specified here becomes a prefix for all the real TP names that are used by the TPs themselves to register. This capability allows TPs to start by

themselves, to be referred to by the generic name specified as TPNAME here, and allows each TP to process a single conversation at a time.

- Default: NO

Step 2. Assembling HCICNFIG

The following shows an example of JCL to assemble HCICNFIG:

Figure 7-10. HCICNFIG ASSEMBLY: JCL to Assemble HCICNFIG.

```
//*
//* PLACE YOUR JOB CARD HERE
//*
//HCICNFIG EXEC PGM=IEV90,PARM='LIST,OBJ,NODECK,TERM,RENT'
//SYSPRINT DD SYSOUT=*
//SYSTERM DD SYSOUT=*
//SYSLIB DD DISP=SHR,DSN=---> THE HCI DISTRIBUTION MACLIB <----
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//SYSLIN DD DSN=&&OBJECT,DISP=(,PASS),UNIT=SYSDA,
// DCB=(RECFM=FB,LRECL=80,BLKSIZE=400),
// SPACE=(TRK,(10,10))
//SYSIN DD DISP=SHR,DSN=---> THE HCICNFIG SOURCE MEMBER <----
//*
//LINK EXEC PGM=HEWL,PARM='LIST,XREF,RENT,REUS,MAP',COND=(0,NE)
//SYSPRINT DD SYSOUT=*
//SYSUT1 DD UNIT=SYSDA,SPACE=(CYL,(10,10))
//SYSLMOD DD DISP=SHR,DSN=---> THE HCI NON-AUTHORIZED LOAD <---
//SYSLIN DD DSN=&&OBJECT,DISP=(OLD,PASS,DELETE)
// DD *
NAME HCICNFIG(R)
/*
```

Step 3. Continue Installation of Your Compuware Product

By completing this chapter, you have finished the installation, maintenance, and customization of all required portions of ECC. Return to the appropriate manual for the Compuware product being installed and continue with the procedure provided there.

Resolving Problems

If any error messages were returned while performing steps in this chapter, refer to the *Enterprise Common Components Messages and Codes* guide.

If you encounter any other difficulties related to the HCI, consult the manuals that were provided with the product invoking the HCI. If problems persist, contact Compuware technical support.

Chapter 8.
HCI Facilities

This chapter explains the configuration settings necessary for the additional features available with Host Communication Interface (HCI) to function with the Compuware products that require it. Currently these include UNIFACE and XPEDITER/DevEnterprise. The sections consist of the following:

- "HCI Features" — Covers internal settings, server activation, journaling, operator communications, dynamic refresh, parallel SYSPLEX, and the VTAM resources.
- "Security and Dispatching" on page 8-20 — Security packages discussed are RACF, ACF2, and Top Secret. Defaults, inbound and outbound conversation requests, system considerations, and dispatching are covered.
- "HCI Diagnosis and Debugging" on page 8-23 — Describes how to capture storage dumps, the HCI journal, stub tracing, and the GTF trace. It also covers using the TCP/IP FTP program.

Note: Abend-AID and Abend-AID for CICS users do NOT need to follow any of these configuration instructions. All of the necessary procedures for these products are contained in their respective documentation.

HCI Features

Internal Password Change

The HCI contains an internal TP that is always available without any configuration parameters required for LU 6.2, and requiring the addition of only one HCICNPCB macro for TCP/IP, which performs all functions that are required to change a user's password.

An example of the TCP/IP HCICNPCB macro is shown below. Only the port number and the name of the TCP/IP region need to be added to this definition. All other operands must remain as shown.

Note: This HCICNPCB macro is required only for TCP/IP communications **without either GCS or UNIFACE**. Password change support for GCS and UNIFACE is automatically generated by the HCI.

Figure 8-1. Example of HCICNPCB for Password Change

```
      HCICNPCB ,                                                    *
            PORT=????,      <-----   SPECIFY DESIRED PORT NUMBER    *
            DATAENC=EBCDIC,          ENCODED IN EBCDIC              *
            LLZZ=(0,2,BINB,INCL,USE,KEEP), SPECIFY LENGTH PARMS     *
            LRECL=512,               MAXIMUM RECORD LENGTH          *
            RECFM=V,                 VARIABLE LENGTH RECORDS        *
            TCPNAME=?????,  <-----   SPECIFY TCP/IP NAME            *
            TPNAME=HCIPSCHG,         SPECIFY PASSWORD CHANGE        *
            TYPE=LOCAL               INDICATE A LOCAL PORT
```

The specifications for this TP came from the IBM manual *Systems Network Architecture - Network Products Formats* - LY43-0081. Although the IBM specifications assume only an LU 6.2 implementation, the HCI makes this function available to TCP/IP clients as well.

Internal APING

Overview

APING and APINGD are LU 6.2 TPs that communicate with each other to prove that connectivity exists between the client machine (running APING) and the server machine (running APINGD). Many SNA providers supply these two programs as part of their application suite. APINGD is an internal TP, which the HCI will attach whenever a conversation request arrives for the TP_NAME HCIAPING. No configuration nor security is required to access HCIAPING from remote clients.

An APING.C program is provided on an AS IS basis by IBM Corporation. For Microsoft SNA Server, a copy can be found on the Backoffice SDK in the \SAMPLES\SNA\APING directory. For IBM Communications Server for NT, it is provided in the base installation. For IBM Communications Manager/2, OS/2 (or OS/2 Access), it is provided in the API support diskette (SNA.ZIP) SNA/CPIC_C/ Directory. As a courtesy, it can be found in the APINGPKG.ZIP at FTP.COMPUWARE.COM in the PUB /UNIFACE /PSVMVS directory.

Invoking HCIAPING

You can invoke the common APING via a command line interface. Below is an example of invoking HCIAPING:

- APING *netid.lu_name* -m LU62B01 -t HCIAPING

  Where *netid* is replaced with the network ID, *lu_name* is replaced with the LU name of an LU owned by the HCI, and mode name and TP name are specified exactly as shown.

  Note: TP_NAME is case sensitive and must be entered in upper case.

Another example shows the addition of a USERID and PASSWORD, which, if present, will be checked for validity by the HCI but will not cause termination of the TP if they are not valid:

- APING netid.lu_name -m LU62B01 -t HCIAPING -u userid -p password

  In this example, a USERID and a PASSWORD have been specified. When the HCI invokes HCIAPING, it will attempt to check the USERID and the PASSWORD. If the attempt is unsuccessful, the MVS console will show the security error messages that describe the error in full. In this way, an installation can check that their security environment is correctly set up at the same time that it checks the LU 6.2 connectivity.

Internal Message Delivery Option

Overview

When both partners of a conversation are serviced on the same HCI address space (could be a SYSPLEX group), VTAM or TCP/IP may optionally be used for communication between them. When VTAM is not used, an internal message delivery mechanism is employed.

Note: The configuration operand, **INTERNAL=,** specifies whether an internal task is to be used for communication between partners on the same HCI (**INTERNAL=YES**) or whether VTAM is to be used for this communication (**INTERNAL=NO**).

This results in the following benefits:

- Significant performance improvement since VTAM is not used.
- Eliminates the requirement to configure the HCI for LU 6.2 communications.
- Eliminates the requirement to update SYS1.VTAMLST and/or SYS1.VTAMLIB.

Enable this feature by specifying INTERNAL=YES on the HCICNGCA macro in the configuration module source. Additionally, when INTERNAL=YES is specified, there is no requirement for any HCICNAMB macros to be coded. Both VTAM and Internal messaging can be enabled within the same HCI.

When a client TP requests a conversation, via CMINIT and CMALLC, the HCI first searches for an active server with the TP_NAME requested by the client. If one is found, and if it is able to accommodate another session, it is assigned this new one, and internal messaging is used. If no TP is currently active, the VTAM ACBs are searched to see if one exists that has the same name as the PARTNER_LU specified in the current SIT entry. If one is found, it is used via internal messaging. If neither of these conditions are true, VTAM is used to established the session between the local HCI and the remote partner.

All CPI-C calls, both standard and extension, are available to TPs using the internal message delivery mechanism. The TP is completely unaware whether VTAM is used or whether its partner is local to the same HCI or remote on some other platform.

Server Activation Facility

Overview

The Server Activation Facility (SAF) is a portion of the HCI that allocates new conversations to existing server applications and starts new server applications whenever appropriate. The SAF is analogous to the Scheduler address space used by APPC/MVS.

Although the functions provided by SAF are similar to the scheduler functions in APPC/MVS, they differ in a number of significant ways. These differences include:

- A single server application can support an unlimited number of conversations simultaneously.
- A “generic” TP_NAME facility allows multiple servers, all with different TP_NAMES, to be known by a single generic name to LUs in the network.
- An “alias” TP_NAME facility allows servers with the same TP_NAME to be known by multiple alias names.
- A server can start under its own control and register itself with the HCI and receive all conversations bound to its TP_NAME.
- An HCI server is bound to an MVS TCB, rather than to an MVS ASCB, thus allowing the HCI to clean up resources held by the server when the TCB terminates rather than having to wait for the address space to terminate.
- Since a TP_NAME must be unique within the HCI (not just unique within the LU_NAME), a server can access network resources via multiple ACBs within the HCI.
- In addition to submitting a batch job or issuing the MVS START command, the HCI can attach a TP program within the HCI address space. This facility is designed for TP programs that need to start quickly and that can manage their own system resources without requiring any JCL allocation facilities.

Control of SAF

Parameters to SAF exist as members in the HCI PARMLIB dataset and as entries in the configuration assembly member. Static parameters define each TP’s execution characteristics as shown below:

- TP_NAME

- An indicator whether this is a generic TP_NAME
- An indicator whether this is an alias TP_NAME
- PARMLIB member name containing initiation instructions
- Maximum number of conversations per instance of TP_NAME
- Maximum number of instances of TP_NAME
- Maximum time job queued but not yet submitted
- Maximum time job submitted but not yet started

From this information, the HCI can submit instances of this server, allocate as many conversations as desired to the existing instance, and can then submit another instance.

If the TP_NAME is generic, it signifies that:

- there will be multiple instances of the server,
- each instance will be pre-started (as opposed to using the SAF submit facility),
- each instance will have the same TP_NAME for the length specified and will have unique values for the rest of the TP_NAME.

As conversation allocation requests arrive, they will contain the same generic TP_NAME. SAF will allocate the conversations to the existing applications limited by the two maximum values contained in the TPT.

Note: For servers that do not require the generic facility capability and do not need to be started automatically, there is no requirement that an entry in the HCI PARMLIB dataset exist for them. The necessary control blocks are created when the application TP registers.

Information needed by the HCI to actually initiate a TP is contained within the member of the PARMLIB dataset named by the MEMBER= operand of the configuration assembly.

Batch Jobs

The HCI will submit JCL to the JES/2 or JES/3 internal reader to initiate TP programs as batch job. JCL within the PARMLIB member is identified by the presence of "//" in the first two positions of the first statement in the member. The first statement is the MVS JOB. The job name is overridden and a unique name is substituted. This unique name is constructed using the job name prefix specified in the HCICNGCA configuration macro, and suffixed to this will be a number to make this name unique and eight characters in length.

SYSPLEX Routed Tasks

Routed task support is part of the SYSPLEX implementation and is always active when SYSPLEX= has been specified via HCICNGCA.

The HCI will issue the MVS ROUTE command to initiate server applications. To use the routed task support, the HCI PARMLIB member named in the HCICNTPT entry must contain at least one record. The word "ROUTE" must start in column one and be followed by either:

- one or more blanks and then the name of the procedure that must exist in one of the system PROCLIBs, or
- a list of SYSPLEX MVS names (separated by commas, enclosed in parenthesis, and followed by one or more spaces) and the name of the procedure that must exist in one of the system PROCLIBs.

Other records are ignored for either of the above.

If no list is specified, the ROUTCMD= configuration parameter is checked. If ROUTCMD=NO was specified, the ROUTE is changed to a simple "START" from "ROUTE". This results in the task being started on the same MVS as the HCI is running

on. If ROUTCMD=YES was specified, all active MVS systems are evaluated and the one currently routed to the fewest number of times from an HCI perspective is selected.

Should HCI SYSPLEX support not be activated, all ROUTE statements will be converted to a simple START. Any route list will be omitted.

ROUTE(System_List)

The route system list will be evaluated as to the number of tasks routed to each MVS system. The one with the fewest number of tasks will be chosen. The SYSPLEX MVS names can be specified as a prefix name (for example, any MVS name having the same prefix becomes part of the list). A prefix name is specified by suffixing an "*" to the prefix (for example, "CW*"). Any MVS name starting with "CW" will be added to the list. This allows the specification of a group of MVS systems with one single list entry. Short notation of a single "*" can be used to denote all MVS systems in the SYSPLEX. You would use this alone with no other names. In addition, it is possible to group entries in the list by using a ";" instead of a comma. The groups are processed from left to right. If all MVS systems within a group are inactive, the next group is evaluated. Again, within a group of one or more active MVS systems, the least routed is selected. Using this grouping technique, it is possible to establish a priority list. For example:

```
(CW03;CW05;CW01)
```

would evaluate from left to right and select the first active MVS system. Whereas:

```
(CW03,CW05,CW01)
```

would select the one with the least number of routes to it. It is possible to mix prefix names and grouping in the list. If the last is a group, it is not necessary to use a ";" prior to the ending parenthesis or "," if not a group.

The system list is denoted by the presence of the left and right parenthesis. This list must follow the ROUTE without spaces. Spaces within the list will be compressed out (not used as if a comma). If none of the MVS systems named are active, it is treated as an error. If the SYSPLEX has been changed to a monosystem, the ROUTE will be changed to a start if the system list contained the MVS system ID on which the HCI is currently running. The HCI will perform the following:

- A unique name will be constructed using the job name prefix specified in the HCICNGCA configuration macro, and to this a number will be suffixed to make this name unique. For all MVS releases prior to 5.0, the unique name will be used as an identifier. For MVS releases of 5.0 and later, the unique name will be used as the value for the reserve START keyword JOBNAME=. This is done to support those sites using Master JCL.
- This unique name will follow the PROC named in the PARMLIB member.
- Following the unique name, the HCI will place the following: ",HCISYS=*xxxx*" where *xxxx* is replaced with the SUBSYSTEM_ID.
- Following HCISYS=, the HCI will place the following: ",HCIUSER=*xxxxxxxx*", where *xxxxxxxx* is replaced with the USERID of the client that initiated the conversation request.
- Following HCIUSER=, the HCI will place the following: ",HCIGRP=*xxxxxxxx*", where *xxxxxxxx* is replaced with the Group name of the client that initiated the conversation request.
- The MVS MCGR macro will be issued with the above command.

Example:

- Assume an HCI PARMLIB member named SYSERVR contains the following:

```
ROUTE(CW01) SRVRPROC
```

- The release of MVS is OS/390 2.5. Therefore the reserve JOBNAME= will be used.
- The HCICNGCA macro specifies a SUBSYSTEM_ID of HCIA, a JOBNAME prefix of HCIJOB, and ROUTCMD=YES.
- A PROC exists in SYS1.PROCLIB named SRVRPROC.
- The HCI will issue the following command:

```
ROUTE CW01,START SRVRPROC,JOBNAME=HCIJOB01,HCISYS=HCIA,
      HCIUSER=*userid,HCIGRP=*group
```

The PROC that the user must have placed in the system PROCLIB has the following requirements:

- The first step in this PROC must execute a program named HCISERVR.
- This program must be passed the parameter string, which is the HCI SUBSYSTEM_ID followed by the Userid, followed by the Group name.
- Subsequent steps in the PROC invoke the user's TP program.

Started Tasks

The HCI will issue the MVS START command to initiate server applications. To use the started task support, the HCI PARMLIB member named in the HCICNTPT entry must contain at least one record. The word "START" must start in column one and be followed by one or more blanks and the name of the procedure that must exist in one of the system PROCLIBs. Other records are ignored. The HCI will perform the following:

- A unique name will be constructed using the job name prefix specified in the HCICNGCA configuration macro, and to this a number will be suffixed to make this name unique. For all MVS releases prior to 5.0, the unique name will be used as an identifier. For MVS releases of 5.0 and later, the unique name will be used as the value for the reserve START keyword JOBNAME=. This is done to support those sites using Master JCL.
- This unique name will follow the PROC named in the PARMLIB member.
- Following the unique name, the HCI will place the following: ",HCISYS=*xxxx*" where *xxxx* is replaced with the SUBSYSTEM_ID.
- The MVS MCGR macro will be issued with the above command.

Example:

- Assume an HCI PARMLIB member named STSERVR contains the following:

```
START SRVRPROC
```

- The release of MVS is prior to 5.0. Therefore the identifier will be used.
- The HCICNGCA macro specifies a SUBSYSTEM_ID of HCIA and a JOBNAME prefix of HCIJOB.
- A PROC exists in SYS1.PROCLIB named SRVRPROC.
- The HCI will issue the following command:

```
START SRVRPROC.HCIJOB01,HCISYS=HCIA
```

The PROC that the user must have placed in the system PROCLIB has the following requirements:

- The first step in this PROC must execute a program named HCISERVR.
- This program must be passed a single parameter that is the HCI SUBSYSTEM_ID.
- Subsequent steps in the PROC will execute the user's server program.

Figure 8-2. Example of a PROC Started Automatically by the HCI

```
//IEFPROC PROC
//HCISTEP EXEC PGM=HCISERVR,PARM=&HCISYS
//STEPLIB   DD DSN=users.authorized.hci.loadlib,DISP=SHR
//USRSTEP EXEC PGM=userpgm
//USRDDS    DD statements as required by userpgm
```

The program, HCISERVR, establishes the security environment appropriately for the execution of the user's server program. That is, the user's program will execute under the USERID of the originating client user at the workstation. This environment matches the one that would exist if the user's program were submitted as a batch job instead of being started as a started task.

If HCISERVR encounters any problems, it abends and the PROC terminates. The HCI will detect that the server has not registered within the allotted time and will notify the client LU that the conversation cannot be started.

HCISERVR will abend with a U0100 abend with a reason code identifying the exact reason for the abend. Abend reason codes from HCISERVR can be found in the *ECC Messages and Codes* guide.

Attached TPs

The HCI can attach TP programs within the HCI address space (no SYSPLEX support). These TPs would be ones that require fast startup and have no need for the allocation services of JCL. In addition, they must be permitted to execute out of an authorized library.

To instruct the HCI to attach a TP, specify the word “ATTACH” in the first position of the first statement in the PARMLIB member for the TP.

Currently only two TP programs use the ATTACH: APING (HCIAPING) and password change (HCIPSCHG).

System Security Concerns

This section covers the implementation of security required for the HCI Server Activation Facility. If this facility is not being used, then the following does not apply. Any references to security package nomenclature will be for RACF. If RACF is not the MVS security being used, the MVS site must make any transpositions to its security package.

The majority of MVS installations use system security. For the SAF process involving submitted batch servers, the security is straightforward — the workstation user must have authorization to perform the following on the MVS system:

- Execute a job with a unique job name as described in “Batch Jobs” on page 8-4.
- Execute the module specified in any job step. In some products this could be the batch initiation of TSO using the IBM module IKJEFT01.

The started task system security concerns are more involved. There are three vitally important pieces of data required by the security:

- USERID
- OWNER
- GROUP

A started task has none of these when the command is issued so they have to be assigned. This is usually done by an exit designated to the security package. Whatever USERID, OWNER, and GROUP is assigned to the started task process, the started task must have

access to the HCI authorized library. This is for the execution of the HCISERVR module in the first step of the procedure. This first step in the started task switches the USERID to that of the user, usually at a workstation, making a request. After this switch, the started task can only access those resources allowed by the workstation user running under the unchanged OWNER and GROUP. Therefore the USERID used by the workstation user must have access to any MVS system and Compuware product libraries designated in those steps following the first step.

The Journal Facility

Overview

Every online system requires a method for recording just what is occurring within the system so that problems can be researched even long after they occurred.

The HCI contains an extensive journaling facility (tracing or logging) that is described below:

The journal facility is a separate task within the HCI. This task is responsible for writing journal records to the current journal dataset. Additionally, this task is responsible for allocating, deallocating, opening, and closing the journal datasets as necessary.

Any number of individual journal datasets can be configured into the HCI although only one of them at a time will ever be allocated and open. Each of these is a VSAM ESDS and would usually be defined with no secondary extents. No pre-formatting of these datasets is required.

One journal is active at a time. The first one specified in the configuration member is dynamically allocated and opened. Journal records are written to this dataset. When it fills, or upon an operator command, this dataset is closed and dynamically deallocated. The next dataset in the list is then allocated and opened, and journaling continues.

When the last journal in the list has been filled, the first one is again allocated and opened. There should be enough journals, each large enough to allow journaling to continue while the batch journal print or journal unload is run, to preserve all journaled data.

Allocating Journal Datasets

The following JCL shows an example of executing IDCAMS to create a journal dataset. The control statements in this example could be replicated in order to create more than one VSAM ESDS.

Figure 8-3. Defining an HCI Journal: Example of JCL to Execute IDCAMS Define

```
//*
//*   place your job card here
//*
//DEFINE  EXEC PGM=IDCAMS,REGION=3M
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
   DELETE HCI.VSAM.JOURNAL CLUSTER
   DEFINE CLUSTER (NAME(HCI.VSAM.JOURNAL)                -
                   TRK(270) VOL(??????)                  -
                   CISZ(16384) SPANNED BUFSP(163840)     -
                   RECSZ(60 32600) NONINDEXED            -
                   REUSE SHR(1 3) SPEED)
/*
```

Journal Contents

The journal dataset can contain records up to 32760 bytes long, although in practice, each record is much shorter. Each record is composed of one or more “GDS Variable” like segments as follows:

A journal segment contains a two-byte length followed by a two-byte segment ID, followed by the text for that segment. One trace record can contain many separate segments, but only those segments that pertain to the event being journaled are written. Listed below are the segment ID's for the journal records:

- X'5000' - Start of Journal Record
- X'5001' - Global Communications Area (GCA)
- X'5002' - Access Method Block (AMB)
- X'5003' - Region Control Block (RCB)
- X'5004' - User Interface Block (UIB)
- X'5005' - Destination Control Block (DCB)
- X'5006' - Conversation Control Block (CCB)
- X'5007' - Free format data
- X'5008' - VTAM Request Parameter List (RPL)
- X'5009' - VTAM Request Parameter List Extension (RPLX)
- X'500A' - Logical Unit Block (LUB)
- X'500B' - TCP/IP Event Area (TEA)
- X'500C' - TCP/IP Port Control Block (PCB)
- X'500D' - TCP/IP Task Status Block (TSB)

Therefore, each journal record will consist of one or more of the segments listed above. Each journal record also contains an event identification that identifies what event is being journaled. This event ID is in the first fullword of the GCA journal record, so it is easy to find. Segment ID's X'5000' and X'5001' exist in each and every journal record, so the GCA is always present.

The batch journal print program can identify each journal event (in English text) and can identify each data area that it is printing. A description of these control blocks is beyond this document. It is only intended for Compuware technical support.

Operator Communications

Overview

Messages to the operator are displayed from the HCI via the standard HCI message processing facility. All error and informational messages are defined in a dataset that is processed by the HCI message processing system. One or more load modules are created that are used to create operator messages. Variable text is provided by the HCI in the calls to display error messages.

Commands can be issued to the HCI via the MVS operator console interface.

Operator Commands

Operator commands are entered to the HCI via a WTOR, or via the MVS MODIFY command depending on the configuration HCICNGCA macro's OPCMD parameter.

Command Syntax

The syntax of the HCI commands is as follows:

```
command operand»,operandOE...
```

The command is separated from the operands by one or more spaces — no comma. If there is more than one operand, each is separated from the next with a comma — no spaces. A command does not have to have operands.

Command List

Below is listed all supported operator commands. The uppercase letters of each command indicate the minimum characters required to accept the short form of the command.

- ACTivate subtask
- DEact subtask
- Display
- Open
- Close
- JMask
- REfresh (modname)
- SDump
- SHUTDOWN
- SWap

Each of these commands is described below:

ACTIVATE Subtask

The ACTIVATE command restarts a VTAM ACB-related subtask that has previously been deactivated either by operator command or by exhausting the retry count in the subtask recovery routine.

DEACT Subtask

The DEACT command directs the HCI to deactivate the subtask named in the command. Operands for this command include normal deactivation and a forced abend deactivation.

DISPLAY

The DISPLAY command allows the operator to display upon his console various components and statuses within the HCI. The possible operands on the display command are:

- ALL — display all HCI status fields
- JOUNRAL — display information concerning the journal
- STOR — display current virtual storage utilization
- CB — display current control block utilization

Other operands will be added as needed.

OPEN

The OPEN command directs the HCI to allocate and open the first journal dataset and begin journaling.

CLOSE

The CLOSE command directs the HCI to close and deallocate the current journal dataset and refrain from any further journaling until an open command is issued.

JMASK

Set the JMASK (Journal Mask) “on” or “off”. “On” causes all journal events to be recorded. “Off” causes no journal events to be recorded. This command takes effect immediately.

- ON — set the journal mask to ones - x‘FFFFFFFFFFFFFFFF’.
- OFF — set the journal mask to zeroes - x‘0000000000000000.

REFRESH (modname)

The REFRESH command directs the HCI to read the configuration module named *modname* from the //FDBDRPL DD concatenation and to dynamically update the running parameters in the HCI from the values in the named configuration module. If *modname* is omitted, *HCICNFIG* is used as the default. See a description of the “Dynamic Refresh Facility” on page 8-12.

SDUMP sysunit,conname

Prepare MVS and the HCI to take an SDUMP. This command sets the four internal values of the HCI as if the operands DMPPFX, DMPVOL, DMPCLAS and DMPUNIT of the HCICNGCA configuration macro had been set to the appropriate values for an SDUMP.

In order for the HCI to take an SDUMP, the DMPUNIT must be set to a valid SYSUNIT name, and the DMPPFX must be set to “SDUMP”. The others, DMPVOL and DMPCLAS, must not have been specified.

This command will reset the DMPVOL and DMPCLAS internal values. There are two positional operands: SYSUNIT and console name separated by a comma (“,”). Each is optional. But if the second is specified, the comma must be present. If the second operand is omitted, those operator commands to set the options for the SDUMP are displayed only. If the second operand is specified, the HCI issues the operator commands to establish the proper SDUMP options for an HCI SDUMP.

The original values can only be reset by restarting the HCI or issuing the REFRESH command. The following show command variations:

- “SDUMP” — displays the operator commands to set the SDUMP options. This form does not change any of the HCI internal values.
- “SDUMP sysunit” — sets the DMPUNIT internal value and displays the operator commands to set the SDUMP options.
- “SDUMP sysunit,conname” — sets the DMPUNIT internal value and issues the console commands to correctly set the SDUMP options.
- “SDUMP ,” — sets “SYSDA” as the internal DMPUNIT name and displays the operator commands to set the SDUMP options.
- “SDUMP *,conname” — uses the currently set DMPUNIT internal value and issues the operator commands to set the SDUMP options.
- “SDUMP ,conname” — sets “SYSDA” as the internal DMPUNIT name and issues the commands.

See “HCI Diagnosis and Debugging” on page 8-23 for more information on storage dumps.

SHUTDOWN

The SHUTDOWN command directs the HCI to terminate. Operands available on this command are “IMMED” for an immediate shutdown, “NORMAL” for a normal shutdown, or “ABEND” for a forced abend situation.

If none of these three operands is entered, “NORMAL” is assumed.

SWAP

The SWAP command directs the HCI to close and deallocate the current journal dataset and allocate and open the next one.

LU 6.2 Session Limits

When the HCI acts as the client or allocating side of an LU 6.2 conversation, there are two ways that the maximum number of sessions, the minimum contention winner and minimum contention loser values can be specified. These three values can be specified in the HCICNSIT entry for the symbolic destination name that is used to allocate the conversations. When any one of these three values is specified, then the HCI uses all three values in its request to VTAM to initiate CNOS processing. If, however, all three of these values are specified as equaling 0 (i.e. MINCL=0, MINCW=0, and MAXS=0), then the HCI allows VTAM to assign these values from the specification on the APPL definition statement in VTAMLST for the HCI application.

Note: Some prior releases of VTAM did not support these values in the APPL definition, so be sure that you are running on a release of VTAM that supports session limit parameters on the APPL statement. If you are not, then you must specify these values in the HCICNSIT entry.

Compuware recommends that VTAM be allowed to negotiate these values by the specification of MINCL=0, MINCW=0, and MAXS=0. In this way, any changes to the CNOS processing in VTAM will be reflected in the HCI execution.

Dynamic Refresh Facility

Overview

The HCI can load a new configuration module and process the parameters contained in this module while the HCI is executing. This section describes the method of creating and refreshing the parameters dynamically.

The Configuration Load Module

Upon startup, the HCI loads a configuration module named HCICNFIG from the //FDBDRPL DD concatenation and extracts its execution parameters from this module. The name can not be overridden.

To prepare for a dynamic refresh, you can update the module named HCICNFIG with new parameters, or you can create an entirely new module with any name you like. The advantage to updating HCICNFIG is that changes become permanent since subsequent executions of the HCI will load this module. The disadvantage is that if there are errors in HCICNFIG a subsequent execution of the HCI may fail if the errors are not corrected before the HCI is stopped and restarted.

When a module named other than HCICNFIG is chosen, either a **complete** or a **delta** module may be assembled. This delta module could contain only the macros that contained parameters you wished to change, where the complete module would have all HCI macros and parameters in it. If a complete module is used, the HCI will ignore any parameters that it cannot refresh dynamically. The advantage to a complete module, is that it can be renamed HCICNFIG and can then be used as the base configuration for the next time the HCI is stopped and restarted. The parameters from the delta module have to be copied into HCICNFIG for them to have an effect on the next HCI startup.

The REFRESH Command

The REFRESH command is an operator command that invokes dynamic refresh processing. This command is entered from the MVS console either as a reply to the

outstanding WTOR or as a MODIFY command depending upon which option is in effect. The format and operands of this command follow:

Using the reply to a WTOR:

- /*nn*REFRESH
- /*nn*REFRESH,modname
  - Where *nn* is the current reply number as assigned by MVS and,
  - The "modname" is an optional parameter that names the new configuration load module, which must reside in the //FDBDRPL DD concatenation. If specified, it must be separated from REFRESH with a comma. If the "modname" operand is omitted, the name HCICNFIG is used.

Using the MODIFY command:

- F *identifier*,REFRESH
- F *identifier*,REFRESH,modname
  - Where *identifier* is the identification (often JOBNAME) of the HCI.
  - The "modname" is an optional parameter that names the new configuration load module, which must reside in the //FDBDRPL DD concatenation. If specified, it must be separated from REFRESH with a comma. If the "modname" operand is omitted, the name HCICNFIG is used.

Upon successful execution of the refresh processing, the HCI will display a complete list of all parameters that were added or changed on the operator console and in the HCI JOB log. You can use the information displayed to ensure that the desired changes were made correctly.

Dynamic Eligible Parameters

The following is a description of the HCI parameter values that can be changed dynamically. If a parameter does not appear in this list it CANNOT be changed dynamically — the HCI must be stopped and restarted before this parameter value will be effective. Unsupported parameters will be ignored without error.

Each of the HCI configuration macros has a "key" associated with it. This key is the parameter that defines the macro in terms of duplicate entries. Following is a list of the HCI configuration macros and what they use as their key:

- HCICNAMB macros — either TCPNAME or ACBNAME (for TCP/IP or LU 6.2 respectively)
- HCICNPCB macros — PORT number
- HCICNJCB macros — both DDNAME and DSNAME
- HCICNSIT macros — SYMDEST
- HCICNTPT macros — TPNAME

The decision whether a macro in a refresh configuration assembly is an update to an existing macro or a new macro is based on the key of the macro itself. If the keys match, an update is attempted. If no macro of the specified type has a key that matches, then the new macro is added.

Table 8-1. Macro Key Operands: Refresh Update or Add Duplicate Entry Determination

MACRO	KEY OPERAND
HCICNAMB	if LUTYPE=TCPIP: TCPNAME=
	if LUTYPE=LU62: ACBNAME=
HCICNPCB	PORT=
HCICNJCB	DDNAME= and DSNAME=

Table 8-1. Macro Key Operands: Refresh Update or Add Duplicate Entry Determination

MACRO	KEY OPERAND
HCICNSIT	SYMDEST=
HCICNTPT	TPNAME=

The values that can be specified for each of these parameters are described in the *HCI Installation Guide*.

- Dynamic eligible parameters for the existing HCICNGCA macro include:
  - DFLTUSR
  - DMPCLAS
  - DMPPFX
  - DMPUNIT
  - DMPVOL
  - JESCHAR
  - JESID
  - JESPRM
  - JESMAIN
  - JRNMASK
  - MAININT
  - OACBINT
  - SECFAC
  - SRVRJOB
- Entire new HCICNAMB macros may be added even if a new protocol is to be supported for the first time.
- No parameters on the existing HCICNAMB macros may be changed.
- Entire new HCICNPCB macros may be added.
- Dynamic eligible parameters for the existing HCICNPCB macros include:
  - TPNAME
- Entire new HCICNJCB macros may be added.
- No parameters on the existing HCICNJCB macros may be changed.
- Entire new HCICNSIT macros may be added.
- Dynamic eligible parameters for the existing HCICNSIT macros include:
  - PORT
  - TCPNAME
  - PLUNAME
  - MODNAME
  - TPNAME
  - MINCW
  - MINCL
  - LLUNAME
  - USERID
  - PASSWORD
  - IPADDR
  - MAXS
- Entire new HCICNTPT macros may be added.
- Dynamic eligible parameters for the existing HCICNTPT macros include:
  - ALIASOF
  - CATEGORY
  - MEMBER
  - MAXCNV
  - RECVLEN

- MAXTPS
- MAXPTIME
- MAXSTIME
- SRVRJOB

Parallel SYSPLEX Support

An MVS SYSPLEX is defined as 1 to 32 MVS images connected together by a hardware (or software) coupling facility (CF). There are numerous options available for a user to provide this CF support. However, these options will not be discussed here. The HCI uses the XCF signaling facility to send and receive messages between components of the HCI when those components are running on different MVS images. Generally, these systems share all DASD including the security system dataset. A JES/2 shared spool environment usually exists. Exceptions to these generalizations can exist. The MVS systems will inter-operate to create the illusion of a single, very big, MVS, and it is this inter-operation that required the HCI to implement SYSPLEX support.

A user who is running an MVS SYSPLEX should have no additional requirements to configure MVS in order to support the HCI.

This section describes the different possible configurations for the HCI in a SYSPLEX environment.

HCI Modes of Operation

HCI Version 2 Release 2 and more recent can run in the execution modes listed below:

- Non-SYSPLEX mode on a Non-SYSPLEX system

  This mode duplicates the functionality of HCI Version 2 Release 1 and earlier.

- Non-SYSPLEX mode on a SYSPLEX system

  This mode duplicates the functionality of HCI Version 2 Release 1 and earlier.

- SYSPLEX mode on a SYSPLEX enabled system

  This mode is new with HCI Version 2 Release 2.

Non-SYSPLEX Operational Mode

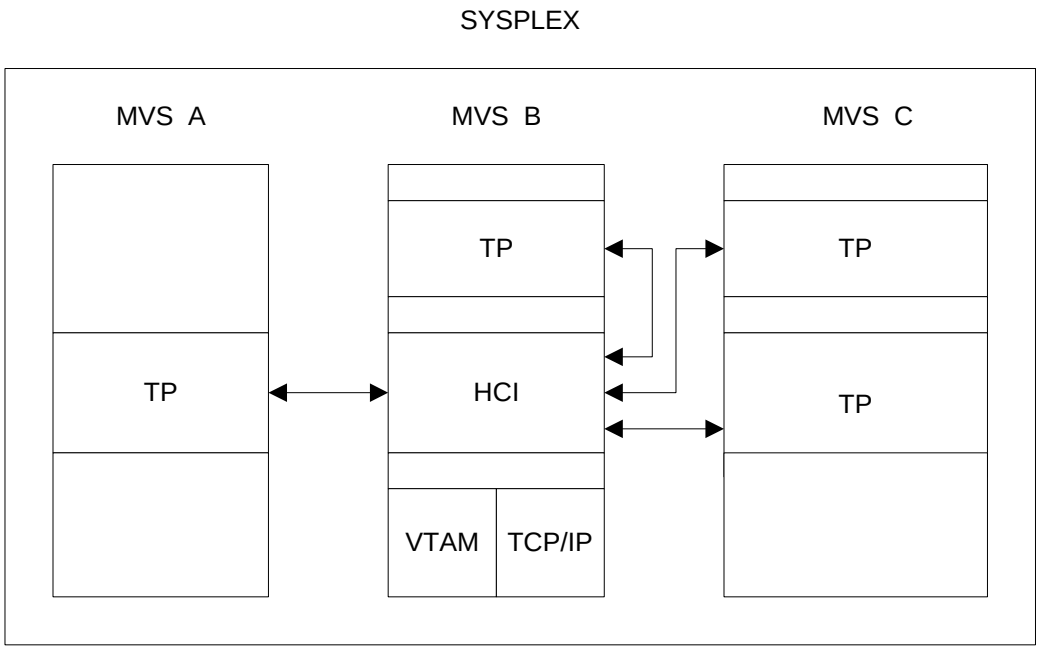
The following paragraphs apply to the Non-SYSPLEX operation mode of execution on either a non-SYSPLEX MVS or a SYSPLEX MVS system.

- This mode requires no HCI configuration macro changes, but it does require that the existing configuration member be reassembled using the HCI 2.2 or more recent macro libraries. Only if other features of this release are desired (e.g. ALIASOF= feature) would changes to the configuration member be required.
- This mode is selected by specifying SYSPLEX=, PREPROC=, and ROUTCMD=NO or by allowing these operands to default to these values by omitting them entirely from the HCICNGCA macro.
- TPs can be submitted (or started) automatically by the HCI, or they may be pre-started by user intervention. These TPs would still use the SUBSYSTEM_ID that is specified on the SYSID= operand of the HCICNGCA macro.
- It is highly recommended that all TPs be re-bound with the V2R2M0 HCISCPIC loadlib/object member, although TPs using HCISTUB do not need to be re-bound.
- The HCI will not invoke any of the new SYSPLEX code, and it can be run on MVS/XA systems (which do not have SYSPLEX support).
- All other new features of this release are available to installations running the HCI in this mode.

SYSPLEX Operational Mode

Prior to HCI 2.2, if a user wanted to run TPs on different MVS systems, he would have to run a separate HCI on each MVS because the HCI could only control TPs that were running on the same MVS image. The HCI's SYSPLEX support eliminates the need to run multiple HCI address spaces. One HCI, running on one MVS image in the SYSPLEX, can control TPs that run on the same or different MVS images. This single HCI provides the capability of running a single TCP/IP stack or VTAM address space (on the MVS that is running the HCI), but it allows TPs to access these communications access methods from any MVS image.

Figure 8-4. Single HCI within a SYSPLEX Servicing Multiple TPs Across Multiple MVS Images



Of course, multiple HCIs' address spaces can co-exist in a SYSPLEX, and each can control TPs anywhere in the SYSPLEX. However, these HCIs do not interact with each other — they remain autonomous entities.

The number of MVS images in the SYSPLEX, and the order in which each is activated, is completely arbitrary and is handled automatically by the HCI. There is no requirement that all of the MVS images be active when the HCI is started. As new MVS images appear in the SYSPLEX, they are automatically added to the HCI, and the preparation job is automatically scheduled to run on these new MVS's. If an MVS becomes inactive, it is automatically deleted from the HCI and is re-added when it becomes active again. The intent is to automate the functions of the HCI as much as possible.

The SYSPLEX mode of operation requires changes to the HCICNGCA macro. Three operands apply to SYSPLEX execution:

- SYSPLEX=
- PREPROC=
- ROUTCMD=

These three operands are discussed in detail later in this document, but essentially:

- The SYSPLEX= operand defines a four-character SUBSYSTEM_ID that is used on all MVS systems in the SYSPLEX. This ID must be different from the SYSID= SUBSYSTEM_ID, and must be unique within the SYSPLEX. It is this SUBSYSTEM_ID

that TPs must use when registering with the HCI. The inclusion of this operand invokes SYSPLEX support in the HCI.

- The PREPROC= operand specifies the name of a procedure (stored in an appropriate PROCLIB) that invokes the HCIYPREP program, which prepares an MVS system for XCF communication with an HCI.
- The ROUTCMD= is only referenced when a ROUTE statement in the HCI PARMLIB has no list.
  - ROUTCMD=YES specifies that the HCI should route MVS START commands to the active member of the SYSPLEX that currently has the least number of routes to it from an HCI perspective.

  or

  - ROUTCMD=NO specifies that the HCI should invoke started task TPs on the local MVS system only.

Installation on a SYSPLEX

All of the steps for installing previous versions of the HCI must still be executed for this version, but additional steps need to be performed:

- Ensure that the HCI load libraries are APF-authorized and available on ALL MVS images. They must be APF-authorized on every MVS, not just on the MVS where the HCI will run.
- A new procedure must be installed in a PROCLIB that is either shared on all MVS images, or the procedure must be installed in a PROCLIB on each MVS image. The PROCLIB must be one from which started tasks can be defined. This procedure invokes the HCIYPREP program that prepares the MVS for HCI SYSPLEX execution. The name of this procedure is up to the user, but whatever name is chosen must be specified on the PREPROC= operand of the HCICNGCA macro. This is not an optional step. See “The PREPROC= Operand” on page 8-18.
- A new four-character SUBSYSTEM_ID must be designated that is different from any other SUBSYSTEM_ID in use on any of the MVS images, and this ID must be specified in the SYSPLEX= operand of the HCICNGCA macro. Just as with the SYSID= SUBSYSTEM_ID, this name must not be included in the IEFSSNxx member of SYS1.PARMLIB.
- Change the word “START” to the word “ROUTE” in any HCI PARMLIB member that defines a TP that can be started on any MVS in the SYSPLEX. In addition, if it can only run on one or more systems, but not all, specify them in a list (see “ROUTE(System_List)” on page 8-5).
- If you want the HCI to automatically start “started task” TPs, you must decide whether these TPs are to be started only on the MVS on which the HCI is running, or whether they can execute on any MVS in the SYSPLEX when only “ROUTE” without a list is specified. Specify ROUTCMD=YES if any MVS can run these TPs. Specify ROUTCMD=NO if TPs must run on the HCI's MVS image.

Note: If ROUTCMD=YES has been specified and the system is not SYSPLEX enabled, it is treated as ROUTCMD=NO.

The SYSPLEX= Operand

Specifying the SYSPLEX= operand on the HCICNGCA macro invokes the SYSPLEX processing functions of the HCI. The value of this operand is a four-character SUBSYSTEM_ID, which must be different from the four-character SUBSYSTEM_ID specified on the SYSID= operand. Like the SYSID= ID, this SUBSYSTEM_ID must be unique among all the MVS images in the SYSPLEX and must not be specified in the IEFSSNxx member of SYS1.PARMLIB.

In the SYSPLEX environment, there are two SUBSYSTEM_IDs that, together, define an HCI. The SYSID= name is the local ID for the HCI (local to the MVS on which the HCI is running). The SYSPLEX= name is the global ID for the HCI. It is recommended that TPs running anywhere in the SYSPLEX use the global ID for registering with the HCI, even TPs that are running on the local MVS image. This is because the HCI automatically converts the global ID to the local ID when it recognizes that the TP is running on the same MVS image as the HCI that controls it. This conversion takes place without user intervention and ensures that TP performance on the local MVS does not suffer any degradation caused by the XCF processing.

Note: It is not possible for a user to run a TP on the local MVS and use the SYSPLEX code in the HCI.

The PREPROC= Operand

This operand specifies the name of the procedure that the HCI will invoke on each MVS image in the SYSPLEX to prepare that MVS for HCI communication. This procedure can be named anything, but whatever name is chosen must be specified in this macro operand. An example of this procedure is shown in Figure 8-5:

Figure 8-5. Example of a Procedure to Invoke HCIYPREP

```
//IEFPROC PROC HCISYSP=DUMY,HCITYPE=
//HCIPREP EXEC PGM=HCIYPREP,PARM='&HCISYSP,&HCITYPE'
//STEPLIB   DD DSN=**** APF-authorized load library ****,DISP=SHR
```

This procedure must be available to every MVS in the SYSPLEX, as the name of a “START” command routed to each MVS by the HCI, and the HCI automatically supplies values for the symbolic parameters that exist in this procedure.

With the exception of specifying the appropriate APF-authorized load library, this procedure should be coded exactly as shown above.

The ROUTCMD= Operand

When “ROUTE” is specified in a PARMLIB member **without** a list:

- ROUTCMD=YES specifies that the HCI is to route MVS START commands to each MVS in the SYSPLEX.

  or

- ROUTCMD=NO specifies that the HCI is to issue MVS START commands only for the local system.

If “ROUTE” is used **with** a list, those specified MVS system names only participate in the selection. If YES is chosen, the HCI will prefix each “START” command with “ROUTE sysname”, which will cause MVS to execute the command on the system designated by sysname. To determine which MVS will run the task, the HCI keeps a count of tasks routed to each MVS. The one with the least number is selected.

The control of which MVS is to execute submitted jobs is left completely up to the installation. Appropriate job classes and initiators must be established by each user to ensure that TP jobs submitted by the HCI run on the appropriate system. When ROUTCMD=YES is specified, JESJPRM=NO should also be specified so that TPs submitted by the HCI are allowed to execute on any MVS image in the SYSPLEX. A complete description of these operands can be found in the *HCI Installation Guide.*

HCIYPREP Program

The HCIYPREP program initializes an MVS such that HCI TPs can run on that MVS and can be controlled by an HCI on another (or the same) MVS image. This program loads executable modules into storage, establishes a subsystem and a MVS/PC routine. When HCIYPREP terminates, these elements remain available to HCI TPs.

HCIYPREP is started automatically on each MVS in the SYSPLEX when:

1. the HCI starts,
2. a new MVS is added to the SYSPLEX, and
3. an old MVS that left the SYSPLEX has subsequently rejoined it.

Under normal circumstances, an operator should never need to intervene and run this program. When HCIYPREP has completed its initialization successfully, it signals this fact to the HCI, which then marks the associated MVS as ready to process TPs. Until HCIYPREP ends normally, the MVS is considered to be in a pending active state, and TPs will not be started on this MVS. A console message indicating which MVS images are in this pending state will be displayed in 2 minutes of the attempted start and then every hour until started. If HCIYPREP did not run successfully, it may have to be started manually after the problem it encountered has been fixed. The operator should issue the same MVS “START” command as the HCI issued, and this command can be found on the MVS SYSLOG.

Shared DASD Considerations

Although the purpose of a SYSPLEX is to have two or more independent MVS images tightly coupled, it does not necessitate total separation. In order to ease maintenance issues, shared DASD should be utilized between the members of the SYSPLEX. One candidate for the shared DASD space should be the HCI authorized loadlib dataset. This is particularly true if the HCI Server Activation Facility routes started tasks to other members of the SYSPLEX. These routed started tasks require access to the HCI authorized library.

VTAM Generic Resources

Overview

VTAM’s Generic Resource facility uses services of the MVS Coupling Facility to maintain a list of *generic* names and the association of each generic name to a *specific* name (APPLID). Remote LUs wishing to establish LU 6.2 or LU 2 sessions with the HCI can do so by specifying the generic name in place of the actual (APPLID) name. When VTAM receives a session request directed to a generic name, it assigns the session to one of the applications that has registered with the generic name. In this way, multiple HCI's can run in a SYSPLEX environment (for example, one on each MVS image), and VTAM will ensure that the session load is balanced among the various HCIs. In addition, each HCI can support multiple LU 6.2 and/or LU 2 ACBs, each of which can register with the same or different generic name. In this environment, multiple LU 6.2 and/or LU 2 sessions can better utilize the CPU resources by being separated onto different ACBs.

Requesting Generic Resource Support

Generic Resource support is requested by adding a new operand to the HCICNAMB configuration macro (or by adding a statement to the AMB configuration source statements). The new macro operand is GENNAME=*xxxxxxxx*, where *xxxxxxxx* is the generic name that the associated ACB is to register with. Multiple HCICNAMB macros can specify the same generic name, but all HCICNAMB macros that do specify the same generic name must be of the same LU type (for example, LU62 or LU2). HCICNAMB macros in different HCIs can specify the same generic name.

Note: ACBNAME, however, must **ALWAYS** be unique within the network.

Operational Considerations

For the HCI to successfully register a generic name, the coupling facility must be installed and must be activated. If it is not, the registration process will fail, a message will be displayed on the master console, and HCI initialization will continue without the generic name. If this occurs, the HCI will request that MVS notify it whenever the status of the coupling facility changes. When this notification occurs, the HCI retries the request to create the generic name. This process continues until the generic name is successfully added. Thus, it is not necessary to stop and restart the HCI for it to begin to honor generic names.

Security and Dispatching

Security Considerations

Overview

User security is handled with the HCI as prescribed by the LU 6.2 architecture and by MVS requirements. The following is a discussion of how user security is supported within the HCI. Note that User Security refers to the process of validating that a particular user is authorized to perform the functions he is requesting. Contrast this with network security that is concerned with encrypting and decrypting data as it flows on the network. Network security is not handled in the HCI.

Supported Security Systems

The HCI supports all three major MVS system security packages:

- RACF
- ACF2
- Top Secret

as well as lesser known security packages.

This is done via the standard MVS system Security Authorization Facility (SAF) interface utilizing the RACROUTE macro. This, of course, permits other security packages to be used including client site developed packages as long as they interface properly to the SAF.

If the security package does not utilize the SAF interface, it is still possible to have the HCI utilize this non-standard package. This is done by coding the necessary access points in the HCI security module (HCIRACRP) that has been provided as a source image.

Note: *After changing HCIRACRP, the client takes full and sole responsibility of its maintenance. Due to the sensitive nature of HCIRACRP, proper precautions should be taken to limit access to its load and source images.*

Default USERID

A default Userid is extracted from the MVS security system (for example, RACF) when any user of HCI registers. Therefore, if the task that is registering was operator started (i.e. not started by the HCI's Server Activation Facility), the Userid that is associated with the task becomes the default Userid. For communications purposes this Userid can be overridden by HCI CPI-C extension calls, or it can be left alone and used as is.

Inbound Conversation Requests

Inbound conversation requests are initiated by the receipt of an FMH-5 (or its equivalent GCS/UNIFACE request message) from the partner LU. Within this FMH-5 can be the following:

- No security Information
- Both a Userid and a Password
- Only a Userid

Note that for LU2 and TCP/IP the HCI constructs an internal dummy FMH-5 that contains the information from whichever type of message initiated the logon processing. This dummy FMH-5 is then used just as the LU 6.2 FMH-5 is.

The HCI performs security validation as follows:

- When no security fields exist within the FMH-5, no security checking is performed. If a user has already registered with the HCI, the conversation is passed to that user, and it executes under his default Userid. If no user is registered, and the Server Activation Facility (SAF) starts an address space, or attaches a TP to handle the incoming request, the default Userid is used, and it is under this Userid that the user program runs. If no default Userid exists, the connection request is denied.
- When both a Userid and a Password exist within the FMH-5, they are checked against the security system. If either is not valid, the conversation request is denied. If both are valid, and a user has already registered, the conversation is passed to him and he continues to execute under his default Userid. If no user has registered, SAF initiates a user address space, and it runs under the Userid that was received in the FMH-5. This Userid becomes the default Userid.
- When only a Userid exists within the FMH-5, the “already verified” bit must also be set, indicating that the user identified by the Userid has entered a valid password, and is recognized as a valid user. The Userid (with no password) is checked against the security system. If the user is not defined to the security system, the conversation request is denied. If the user is defined, and has already registered with the HCI, the conversation is passed to him. If no user has registered for the conversation, SAF initiates a new address space and it executes under the Userid that was received in the FMH-5. This Userid becomes the default Userid. The JCL initiating the task can contain USER and PASSWORD operands that will override the default Userid.
- At any time after accepting the conversation, the user may extract the Userid and Password (if any) via the CPI-C extension security calls. A Password will only be returned if one was received on the incoming FMH-5 that initiated the conversation or if one was set by the user prior to issuing the extract call.

Outbound Conversation Requests

Outbound conversations are initiated by the user issuing CMINIT and CMALLC calls, which cause the HCI to send an FMH-5 to the partner LU. It is the current setting of SECURITY_TYPE that dictates how security fields are set into the FMH-5. If SECURITY_NONE is set, no Userid nor password are placed into the FMH-5. If SECURITY_SAME is set, only Userid is placed in the FMH-5, and the “already verified” bit is set. If SECURITY_PROGRAM is set, both Userid and password are placed in the FMH-5, and the “already verified” bit is not set. Note that the default Userid will be used unless overridden by the user prior to issuing the CMALLC call. Note, also, that the Userid and password values CANNOT be changed unless the user has previously set SECURITY_PROGRAM, overriding the default, which is SECURITY_SAME.

TP_NAME Security

An optional feature of the HCI is to require that all userids belonging to transaction application programs be defined to the security system as having access to the TP_NAMES that they are registering as. This facility is a protection against rogue

application programs masquerading as valid programs and intercepting transactions that were not meant for them to process. The facility involves the security administrator defining an entity, which is the HCI SUBSYSTEM_ID followed by a period followed by the TP_NAME, under the security class of FACILITY. Then, the administrator must grant "READ" access to any USERID that is to have the authority to run this application. This facility is available on a TP_NAME by TP_NAME basis, because if the HCI finds that the TP_NAME in question has not been defined as an entity to the security system, it allows unlimited access to the TP_NAME. Therefore, an installation can choose to protect some of the local TP_NAMES, but not others.

NO Security

The majority of MVS installations utilize system security. This could be RACF, ACF2, Top Secret or other. However, a few installations do not have system security at all. It will be necessary to confirm the following for the HCI configuration:

- TPSEC=NO has been specified in the HCICNGCA definition.
- USERSEC=NO has been specified in all the HCICNTPT definitions.
- The module HCIRACRP has had the OPCODE in the statement following the statement labeled MAIN0000 changed from "NOP" to "B". This will effectively bypass all security references.

System Security for the HCI

In order for the HCI to properly function, it must be permitted access to the following regardless of whether it is a started task or submitted job:

- Authorized load library containing the HCI modules.
- HCI PARMLIB dataset containing HCI Server Activation Facility members.
- Load library PDS containing the HCI configuration module.

The HCI must be authorized for the following via the MVS system security package:

- To generate SDUMPS, if DMPPFX=SDUMP has been specified (the SDUMP specification is recommended by Compuware),
- Output submitted jobs via an internal reader.
- For operator commands with "update" authority:
  - START (if any TP will be initiated via the HCI START parm)
  - STOP
  - CANCEL
  - MODIFY (if OPCMD=MODIFY has been specified)
  - ROUTE (if SYSPLEX=xxxx has been specified)

For MVS system security regarding the Server Activation Facility of the HCI, see "System Security Concerns" on page 8-7 and "Control of SAF" on page 8-3.

Dispatching Considerations

It can be difficult to determine the "pecking" order of task execution in an MVS system without knowing the functions of the tasks. The HCI is much like a multiple user region because it services many other regions much like VTAM, but it uses resources like VTAM and VSAM. Its position in the pecking order (dispatching priority) should be set so that it is below the subsystems VTAM, VSAM, and communication subsystems, (for example, TCPIP or TCPAccess by InterLink) but above the regions (TP applications) that it services, namely the regions started via the HCI Server Activation Facility.

If the HCI and the TP applications are all started tasks and are all lumped together in one dispatching group, it may be necessary to watch closely for any bottlenecks involving the

HCI. If one is found, it would then be necessary to have the TP application started tasks run at a lesser dispatching order than the HCI.

The TP applications should run at a dispatching priority pursuant to that of any other on-line application (for example, TSO). This would then provide adequate response time to the workstation users.

HCI Diagnosis and Debugging

Overview

In case of problems with the HCI, Compuware may ask you to provide some or all of the following information as an aid in diagnosing HCI problems:

- Storage dumps
- HCI journal
- Stub tracing
- GTF trace

Each of these items is discussed in detail below:

Storage Dumps

The HCI invokes MVS dumping facilities whenever it detects a problem that cannot automatically be corrected. Two major types of dumps can be requested by the HCI:

- SNAP dumps
- SDUMP dumps

SNAP dumps can be directed to a disk dataset or to SYSOUT, based on the DMPCLAS, DMPPFX, DMPUNIT, and DMPVOL operands on the HCICNGCA macro. Specifying DMPCLAS obviates the need for the other DMPxxx operands. Specifying DMPPFX, DMPUNIT and DMPVOL obviates the need for DMPCLAS. SNAP dumps are relatively slow to generate, and, as such, execution of the HCI may be suspended for an unacceptable length of time.

SDUMP dumps are always directed to the installation defined SYS1.DUMPxx (or equivalent) datasets. These dumps can be managed by the installation more easily than SNAP dumps, and they can be copied easily to disk or tape. These dumps are generated in a very short length of time, thus suspending HCI execution for a minimal duration. SDUMP dumps are requested by coding DMPPFX=SDUMP (and omitting DMPCLAS and DMPVOL operands) on the HCICNGCA macro. **Compuware strongly recommends that DMPPFX=SDUMP be coded**.

Any dumps sent to Compuware should contain the following:

- Abending PSW.
- Abending register contents.
- Storage: region, private, common, LPA, SQA and LSQA.
- Region system control blocks, e.g., ASCB, TCBs, IRBs, etc.
- Global resource enqueues.
- System trace tables.
- Dump summary.

If you are generating an SDUMP, then specify the following options:

```
RGN,PSA,CSA,LPA,SQA,LSQA,GRSQ,TRT,SUM
```

and

```
TYPE=XMEME
```

If the HCI is running in a SYSPLEX, and one or more of the TP applications are running on a different MVS image than the HCI, then also specify the following option:

```
COUPLE
```

Also set the SDUMP HCI operator command (see “SDUMP sysunit,conname” on page 8-11).

HCI Journal

The journal provides the most comprehensive debugging aid available. Although not required for most HCI execution, when a problem exists the journal should be created that contains the execution error. The journal facility is always available in the HCI, but based on the setting of the journal mask operand in the HCICNGCA macro and the availability of journal VSAM datasets, the journal may or may not be active at any one time.

The journal mask can be changed dynamically with the HCI operator console command JMASK (see “JMASK” on page 8-11). New journal datasets can be added and activated without requiring an HCI refresh. See “The Journal Facility” on page 8-8 for a complete discussion of the journal datasets and the operator commands available to control the use of these datasets.

If it is necessary to provide Compuware with the journal data, it should be prepared for either FTP transmission or magnetic tape media. Using FTP transmission is the preferred method - it saves time. Never should a print image of the journal data be sent to Compuware for analysis - this will delay problem analysis. For FTP transmission, see “Preparing Journal Data for FTP Transmission” on page 8-24 and “Using TCP/IP FTP Program” on page 8-28. If sending magnetic tape see “Offloading Journal Data for Magnetic Tape” on page 8-25.

Preparing Journal Data for FTP Transmission

The contents of a journal dataset can be reformatted into a dataset that is appropriate for FTP transmission to Compuware's FTP site. An example of the JCL to perform this reformat is shown below:

Figure 8-6. Journal Unload JCL: Example of JCL to Unload a Journal Dataset

```
//*
//*  PLACE YOUR JOB CARD HERE
//*
//UNLOAD  EXEC PGM=HCIJUNLD
//STEPLIB   DD DSN=users.authorized.hci.loadlib,DISP=SHR
//HCIINDD   DD DSN=HCI.JOURNAL.DATA,DISP=SHR
//HCIOUTDD  DD DSN=users.new.journal,DISP=(,CATLG),
//             UNIT=SYSDA,SPACE=(TRK,(15,15),RLSE,CONTIG)
//SYSUDUMP  DD SYSOUT=*
//ABNLIGNR  DD DUMMY
```

After the data has been successfully unloaded, the unloaded dataset can be transmitted via FTP to Compuware's FTP site. See “Using TCP/IP FTP Program” on page 8-28.

If the HCIJUNLD program is not available in the HCI load library, it is available on the Compuware FTP site in the PUB/HCI/FAQ directory in the HCIJUNLD.OBJ module. This module contains JCL and object modules for a link and go execution. It will be necessary to supply some parameter data pertaining to the site of execution. Don't forget to use the BIN option in FTP since this module contains binary data.

Offloading Journal Data for Magnetic Tape

If the TCP/IP File Transmission Program is not available, the journal data can be sent on magnetic tape to Compuware. Use the journal unload JCL with appropriate tape unit parameters for the HCIOUTDD DD statement. No DCB parameters need to be specified.

If the HCIJUNLD program is not available, then the IDCAMS REPRO function can be used to unload the journal to magnetic tape. Regardless of the technique used, inform Compuware of the technique used. This will reduce wasted time by the Compuware support staff and help in a quick resolution to the problem.

Printing the Journal Data

Never send a print image of the journal data to Compuware to be analyzed - **this will delay problem analysis**. However should it become necessary to print the journal data, the following JCL can be used to minimally format it.

Figure 8-7. Journal Data Printing: Example of JCL to Print the Journal Dataset

```
//*
//*  PLACE YOUR JOB CARD HERE
//*
//JRNLPRT EXEC PGM=HCIJRPRT,REGION=32M
//STEPLIB   DD DSN=users.authorized.hci.loadlib,DISP=SHR
//*
//LOGIN     DD DSN=HCI.JOURNAL.DATA,DISP=SHR
//SYSPRINT  DD SYSOUT=*
//LOGPRINT  DD SYSOUT=*
//SYSUDUMP  DD SYSOUT=*
//ABNLIGNR  DD DUMMY
```

HCISCPIC (STUB) Tracing

The HCI Journal contains all operational information about a request after it passes the interface (HCI subsystem and HCISCPIC). However, it is possible to have a problem that is not recorded in the journal but originates in the interface. In order to help diagnose these problems, a trace facility has been added to HCISCPIC. The stub tracing is activated via the presence of a “dummy” HCITRACE DD statement in the execution step of the TP program:

```
//HCITRACE DD DUMMY
```

Due to the method of interface, it is not possible to use a dataset to capture the trace information. Instead the trace information is written to the programmer. This could result in many entries being written to the MVS system log. Thus when this tracing is used, the problem re-creation should be as short and concise as possible.

Since there is no raw data for this, it will be necessary to provide a print image of the job output from JES or the section of console log containing the trace information.

Note: *This method of diagnosis should be used only when absolutely needed. When not requiring the trace, do not include the dummy HCITRACE DD statement in the JCL.*

GTF Trace

There are two instances when a GTF trace may be required. One, is for a VTAM trace. The other is for an HCI SYSPLEX trace. All aspects of preparing, starting and stopping the GTF as well as sending the GTF data remains the same. The only difference is the GTF

parameters. The VTAM tracing parameters can be found in Figure 8-11 on page 8-27. The HCI SYSPLEX tracing can be found in Figure 8-10 on page 8-27.

Note: In an active SYSPLEX environment, it is not possible to use the HCI journal facility to record the execution of the MVS systems other than the local MVS that contains the HCI itself. Because some sort of journal is required, the MVS GTF Trace facility has been chosen to record SYSPLEX processing.

CAUTION:

Never send a print image of the GTF data to Compuware to be analyzed - this will delay problem analysis. If the GTF data has already been collected, see "Preparing GTF Data for FTP Transmission" on page 8-28 or "Offloading GTF Data for Magnetic Tape" on page 8-28. If you need help in gathering the GTF data, read the following sections.

Preparing GTF to Gather HCI Data

The following steps should be followed to prepare to use GTF to gather HCI data:

- Allocate a dataset to contain the GTF trace output.
- Create a GTF procedure to invoke GTF with the dataset.
- Create a SYS1.PARMLIB member to contain the GTF execution time parameters.

Below is an example of a job that allocates a dataset to receive the GTF trace data:

Figure 8-8. Allocate GTF Dataset: Example of Allocating a GTF Dataset

```
//*
//*  PLACE YOUR JOB CARD HERE
//*
//ALLOC   EXEC PGM=IEFBR14
//GTFDATA   DD DSN=HCI.GTFTRACE.DATA,DISP=(,CATLG),UNIT=SYSDA,
//             SPACE=(CYL,(4),,CONTIG),VOL=SER=volser,
//             DCB=(RECFM=VB,LRECL=8232,BLKSIZE=8236,DSORG=PS)
```

Each installation should choose an appropriate dataset name and volume serial number for this dataset. Code the DCB parameters exactly as shown.

Below is an example of a procedure to invoke GTF:

Figure 8-9. Example of a GTF Procedure.

```
//GTFHCI  PROC MEMBER=GTFHCI
//IEFPROC EXEC PGM=AHLGTF,PARM='MODE=EXT,DEBUG=NO,TIME=YES',
//             REGION=2280K,DPRTY=(15,15)
//IEFRDER DD DSNAME=HCI.GTFTRACE.DATA,DISP=SHR
//SYSLIB  DD DSNAME=SYS1.PARMLIB(&MEMBER),DISP=SHR
```

Associated with the GTF procedure, is a SYS1.PARMLIB member. Examples of this member are shown below:

Figure 8-10. Example of SYS1.PARMLIB GTF Member for GTF HCI SYSPLEX Tracing

```
TRACE=USRP
USR=(333)
END
```

Figure 8-11. Example of SYS1.PARMLIB GTF Member for VTAM Tracing

```
TRACE=USRP
USR=(FE1,FE2,FE4,FEF,FF0,FF1,FF2)
END
```

Starting GTF

The GTF procedure is started by entering a command on the system console or from an SDSF panel. An example of this command is shown in Figure 8-12.

Figure 8-12. Example of a System Command to Start GTF

```
START GTFHCI.GTF

The GTF program will display startup information, including
a listing of the parameters that are to be used for this
invocation. If its initialization is successful, GTF will
request that the operator confirm the startup parameters
by replying to an outstanding WTOR. This message is shown
below:

*nn AHL125A RESPECIFY TRACE OPTIONS OR REPLY U
Using the reply number nn, the operator should enter the
following on the system console or on an SDSF panel:

Rnn,U

GTF will display more diagnostic information, finished by
the following message:

AHL031I GTF INITIALIZATION COMPLETE
```

Note that when GTF has successfully been started, actual tracing to the GTF dataset will not occur until the next TP registers with the HCI. Thus, in-flight conversations will not be traced. It is a good idea to start GTF before starting any HCI TPs so that all information concerning those TPs will be traced.

Stopping GTF

GTF is stopped by entering the following command on the system console or from an SDSF panel:

Figure 8-13. Example of a System Command to Stop GTF

```
P GTF

GTF will respond with the following message:

AHL006I GTF ACKNOWLEDGES STOP COMMAND

Then MVS will report that GTFHCI has ended
```

Sending GTF Data for Analysis

The contents of the GTF dataset can be reformatted into a dataset that is appropriate for FTP transmission to Compuware's FTP site or placed onto magnetic tape to be mailed. Refer to the applicable sections below.

Preparing GTF Data for FTP Transmission

In order to send the GTF data via FTP, it will be necessary to reformat the data. An example of the JCL to perform this reformat is shown below:

Figure 8-14. Example of JCL to Unload the GTF Dataset

```
//*
//*  PLACE YOUR JOB CARD HERE
//*
//UNLOAD  EXEC PGM=HCIGUNLD
//STEPLIB   DD DSN=users.authorized.hci.loadlib,DISP=SHR
//HCIINDD   DD DSNE=HCI.GTFTRACE.DATA,DISP=SHR
//HCIOUTDD  DD DSN=users.new.gtfdata,DISP=(,CATLG),
//             UNIT=SYSDA,SPACE=(TRK,(15,15),RLSE,CONTIG)
//SYSUDUMP  DD SYSOUT=*
//ABNLIGNR  DD DUMMY
```

After the GTF data has been successfully unloaded, the unloaded dataset can be transmitted via FTP to Compuware's FTP site. See “Using TCP/IP FTP Program” on page 8-28 for how to send via FTP.

Offloading GTF Data for Magnetic Tape

If the TCP/IP file transmission facility is not available, the GTF data can be sent on magnetic tape to Compuware. Use the GTF unload JCL with appropriate tape unit parameters for the HCIOUTDD DD statement. No DCB parameters need to be specified.

Using TCP/IP FTP Program

Introduction

A part of TCP/IP implementation is a file transmission facility called File Transfer Protocol (FTP). Although FTP has restrictions, it remains a valuable tool for sending

information quickly from one site to another. The restriction that must be circumvented is its inability to send variable length (or undefined) binary format records in such a way that the original records can be reconstructed at the receiving site. Because of this, only fixed length records that contain binary data can be transmitted. The programs HCIJUNLD and HCIGUNLD are designed to read the variable length input data, and create a fixed length output dataset that can be transmitted by FTP. The format of the records created by HCIJUNLD and HCIGUNLD are known by the corresponding programs that re-create the original datasets.

Invoking FTP

If TCP/IP is available on an MVS host, then FTP can be invoked from a TSO session with or without ISPF. If TCP/IP is not available, then the output of HCIJUNLD or HCIGUNLD must be transmitted to a workstation that does have TCP/IP on it before the data can be sent to Compuware.

The remainder of this section assumes FTP is available on the host.

The initial commands to start the FTP client program and to access Compuware's FTP site are not shown since each customer can have (or not have) a firewall site that imposes restrictions on what can be sent through the Internet and dictates the format of the commands to initially access Internet sites.

Figure 8-15. Invoking FTP: Example of an FTP Session with Compuware

```
Assuming a host-based TCP/IP, and no firewall, the user would
enter the following command on any ISPF panel:

--->  TSO FTP FTP.COMPUWARE.COM
220 WWW Windows NT FTP server
Command:

--->  user anonymous
>>>User anonymous
331 anonymous access allowed, send identity (e-mail name)
    as password.
Password:

--->  *** enter your e-mail name ***
>>>Pass ********
230 anonymous user logged in as anonymous (guest access)
Command:

--->  cd pub/hci/incoming
>>>CWD pub/hci/incoming
250 CWD command successful
Command:

--->  mkdir *** enter your customer name ***
>>>MKD * customer_name *
257 MKD command successful
Command:

--->  cd * customer_name *
>>>CWD * customer_name
230 command successful
Command:

--->  bin
>>>BIN
200 Type set to 1
Command:

--->  put users.new.gtfdata (or users.new.journal)
>>>PUT
226 Transfer Complete
Command:

--->  quit
```

Notes on FTP example:

After you have successfully changed the directory to PUB/HCI/INCOMING, and you attempt to create a new directory with your company name, you might get a reject message saying that you "...cannot create a file when that file already exists". This message indicates that the directory you are trying to create has been created already. Ignore the message, and issue the CD command to enter this directory.

Note: Be sure to set the transfer type to BIN.

Storage Estimates

The storage utilized by the HCI is very use oriented. The amount of storage that is being used can be categorized into four types each above and below the line:

1. Private
2. Common
3. Module
4. Buffer

Private and common storage can each be easily more than 1 MB above the line. The storage used below the line for private and common is minimal at around 20k. Module storage is under 700k, the majority of it above the line - less than 50k below the line. While buffer storage can be less than 100k above the line. Above the line storage is always used unless unavailable or not permitted to be used by some of the very few MVS facilities.

These are all rough estimates and the actual storage size is directly related to the functions the HCI is performing. The actual storage utilization can be displayed on the operator console using the "DISPLAY STOR" HCI command.

To help explain where some of the storage is used, Table 8-2 shows those control blocks whose quantity is user controlled:

Table 8-2. Control Block Sizes: Storage Orientation of Pertinent Control Blocks

Control Block	Description	Size (decimal)	Storage Location	Above/Below 16M line
AMB	Access Method Block	728	CSA	below
CCB	Conversation Control Block	1264	Private	above
DCB	Destination Control Block	760	Private	above
JCB	Journal Control Block	104	Private	above
PCB	TCP/IP Port Control Block	488	Private	above
RCB	Region Control Block	96	CSA	above
TPT	T P Profile Table	320	Private	above
UIB	User Interface Block	1040	CSA	above
WRE	Work Request Element	256	CSA	above

The WREs are usually the largest storage item. Currently the HCI is intolerant of running out of most internal control blocks. Always provide a few extra of each where possible and monitor storage utilization via the TSO/ISPF HCILOOK facility.

CPU Cycle Usage

The HCI has been developed to use a minimum amount of execution time but allowing the maximum amount of flexibility. During development, CPU execution time monitors were used to determine any code that was using excessive time. Any code found using excessive time was consequently changed to use less CPU time. The HCI is very frugal in the amount of CPU time it uses.

Glossary

Abend-AID. The Compuware product for fault diagnosis that provides an immediate, detailed analysis of application program failures in a comprehensive report format.

Abend-AID for CICS. The Compuware product for fault diagnosis that provides a full range of analysis functions for managing abnormal CICS transaction terminations (abends) and CICS region outages.

Abend-AID products. A generic name for the Abend-AID product family.

Abend-AID Report. (1) A set of records containing data extracted at abend time from the affected MVS region and associated control blocks that are stored in the Abend-AID DDIO report file. (2) A readable report produced by CWDDSUTL at view time using the records stored in the Abend-AID DDIO report file.

Abend time. The time when an abend occurs and Abend-AID products perform their analyses.

Allocation group. A set of blocks assigned to a DDIO file member when the member requires additional storage.

APF Authorization. APF can be used to restrict access to system functions and can require that all modules loaded by an authorized program be loaded from an authorized load library.All modules loaded by LMSINIT must be loaded from authorized load libraries or LMSINIT will fail. Only individuals with proper RACF authority will have permission to update authorized load libraries.

Assembler Language Processor. One of four language processors provided by Compuware. This language processor accepts Assembler output, builds sort work records, sorts and merges the records, and merges the records with the listing to produce Processor control blocks that can then be used as input to other Compuware products.

Automatic lock. Automatic locks are created whenever a member is added to a DDIO file format using the AUTODELETE=DUPS or AUTODELETE=STAGED option. The most recent member is automatically locked. See also **manual lock**.

AUTODELETE. An attribute of a DDIO file, specified during formatting of the file, that determines the action to be taken when the file becomes full and an attempt is made to add a new member to the file.

Base Services. Provides basic viewing server-related services. It enables Distributed Viewing Support.

Batch File Utility. A CSS utility used to prepare and manipulate DDIO files.

CEC. Central Electronic Complex designates the physical computer which can contain one or more LPARs. CECs can be of the older type designated as G1 through G6 or can be the new z/800 or z/900.

CEC Licensing. The number of MSUs provided by that CEC across all LPARs.

CGI. Common Gateway Interface. A way of interfacing computer programs with HTTP or WWW servers, so that a server can offer interactive sites instead of just static text and images.

CGI-bin. Common Gateway Interface-BINaries. A special directory where common gateway interface (CGI) scripts are kept.

C Language Processor. One of four language processors provided by Compuware. This language processor accepts C compiler output, builds sort work records, sorts and merges the records, and merges the records with the listing to produce processor control blocks that can then be used as input to other Compuware products.

COBOL Language Processor. One of four language processors provided by Compuware. This language processor accepts COBOL compiler output, builds sort work records, sorts and merges the records, and merges the records with the listing to produce processor control blocks that can then be used as input to other Compuware products.

Compuware Shared Services (CSS). A set of components used by several Compuware products to provide storage, retrieval, and maintenance for source listings and diagnostic reports.

Compuware Viewing Facility (Compuware/VF). The function of CWDDSUTL that allows the viewing, printing, locking, unlocking, and deleting of DDIO file members online.

Customer Modification Facility (CMF). This facility allows you to migrate from a previous release of CSS to CSS 7.4 or higher if you are using

any site-restricted zaps. This facility can retrofit your zaps to the SMP/E environment.

Customized Translation Table. An optional table provided by Compuware customers for translating non-printable characters. It is used by the batch file utility and Compuware/VF for printing or displaying lines containing storage displayed by Abend-AID products at abend time.

CWAASDUT. The batch file utility used for Abend-AID shared directories and report databases. CWAASDUT allows you to manage the files used by an Abend-AID shared directory and its associated abend report files. The shared directory is used to process all directory requests related to an abend report file and maintain information about database activity.

CWASSECU. The name of the user-coded Security Exit program.

CWDDLPUT. The batch file utility program used to manage source shared directories and databases for all of the supported languages. The shared directory maintains information about source listing processing and database activity. XPEDITER/TSO (releases 7.0, 7.1, and 7.2 only) supports source shared directory processing for C language, VisualAge PL/I, and Enterprise PL/I if using LONGNAME compiler support. XPEDITER/CICS (releases 7.4 and 7.5 as well as XPEDITER/TSO releases 7.0, 7.1, and 7.2) supports the use of source databases for all languages as standalone DDIO files only.

CWDDSUTL. The program for Compuware/VF and the batch file utility used by Abend-AID, XPEDITER/CICS, and XPEDITER/TSO. CWDDSUTL allows you to print, delete, lock, unlock, copy, move, import, and export DDIO file. members, and to list DDIO file directory entries. It also allows you to initially format a DDIO file

CWFXSDUT. The batch file utility used by Abend-AID for CICS. CWFXSDUT allows you to manage the files used by a shared directory and its associated transaction databases. The shared directory is used to process all directory requests related to a transaction database.

CWPPRMO. A language processor input dataset that contains all of the language processor options.

DBMODEL. The name of an OBJECT used to define all of the specifications needed to define a new source listing database during "dynamic database creation." The information is stored in the Source Listing Shared Directory specified. It is used to build the dynamic source listing database define and format functions when the dynamic database creation process is invoked during the C Language Processor source member create.

DDIO (Dump Dataset Input Output). A Compuware proprietary file access method. The CSS language processors analyze the output from a compiler or assembler execution and store that information in a source listing DDIO file. CSS DDIO files are used to store diagnostic reports, transaction reports, and source listings. The term **DDIO file** is a generic term used to refer to datasets that store the reports and listings from Compuware products that use CSS.

DDIO File. A generic name for a report file or source listing file.

DDIO File Member. A generic name for an diagnostic report in a report file, a source listing in a source listing file, or a transaction report in a transaction database.

Directory entry. A record in the fixed portion of the DDIO file (Directory) that contains information specific to a member. Each member has a corresponding directory entry. The number of directory entries is specified during formatting of the DDIO file and cannot be changed without reformatting the file.

DIRX Report. A report produced by the DIRX command of the batch file utility.

Distributed Viewing Support (DVS). Allows Abend-AID (version 9.0 and more recent) Abend-AID for CICS (version 4.4 and more recent) users to view both merged and base reports, where either the base report or listing file resides on a remote system. Abend-AID users can have source support across remote MVS systems that don't share DASD.

DYNCREATE. A term used to denote the "dynamic database creation" process that can be invoked when using the C Language Processor shared directory and source listing databases for its compiler output (CWPDDIO). A DBMODEL must first be defined in the desired Source Listing Shared Directory using the batch utility CWDDLPUT with DYNCREATE=YES specified to allow invoking of the dynamic database creation process. The CWPDDIO must specify the Source Listing Shared Directory DSN name. If not enough space is found in any of the attached source listing databases, and DYNCREATE=YES, then the dynamic database creation process is invoked, the new source listing database is created and attached, and the new source member is created.

Enhanced Listing. A convenient source of quick reference information and program documentation that merges DMAP and CLIST information, in

addition to error and diagnostic messages, with a COBOL source listing.

Enterprise Common Components (ECC). ECC is the packaging method for the following Compuware facilities:

- Compuware Shared Services (CSS)
- License Management System (LMS)
- Base Services
- Host Communications Interface (HCI)

Entry. A generic name for a transaction diagnostic report in a transaction database for Abend-AID for CICS.

Formatting a DDIO File. The preparation of an allocated file to be used as a DDIO file.

Full DDIO File. A DDIO file for which all allocation groups and/or all directory entries are allocated to members.

Host Communications Interface (HCI). A facility that provides connectivity between mainframe-based programmer productivity software and peer node software running on other platforms in a network. It allows application programs to communicate over any one of several protocols without knowledge of which protocol is in use at any given time.

Installation JCL Customization Facility. A CSS facility consisting of a set of screens that prompt you to enter installation information. This information is used to build the jobs necessary to perform the CSS SMP/E installation.

Language Processor (LP). A processor that converts assembler or compiler output into input for other Compuware products.

License Administration Utility (LAU). The LAU is the license administration control center for your organization's IT professional who is responsible for managing your access to Compuware products. The LAU is an ISPF application that enables

- the creation of a License File
- the import of License Certificates into a License File
- the maintenance and export of a License File
- the reporting and analysis of your License File and License Certificate information in virtual storage

Your organization's License Administrator will also set up License Management System parameters and system operation options using the LAU.

License Certificate. English-like readable electronic records that contain a portion of the information from your license agreement for a product release's use at a particular site. The License Certificate is used to update your License File.

License File. Dataset containing imported License Certificate information for all licensed releases of Compuware products. This file is in ASCII text format.

License Management System (LMS). Facility that enables you to centrally administer Compuware product License Certificates and manage access to Compuware products at your site. The LMS includes several components that together enable you to establish, maintain, diagnose, and upgrade access to the Compuware products licensed by your enterprise. The LMS replaces the customer profile utility provided with earlier versions of Compuware mainframe products.

Line command. A command entered next to the line to be processed. A line command is executed only for the specified member.

LMVERIFY. See **Verification Report**.

Locked member. A DDIO file member that was manually locked using the batch file utility or Compuware/VF, or automatically locked as a result of AUTODELETE=DUPS or AUTODELETE=STAGED processing. A manually locked member is identified by an M in the directory report or directory screen while an automatically locked member is identified by an L in the directory report or directory screen. When manually locked, a member cannot be automatically deleted when the DDIO file becomes full.

LPAR. Logical Partition is one operating system image, such as OS/390, z/OS, or z/OS-e.

LPAR Licensing. Refers to the defined capacity or "capped capacity" of the LPAR itself.

Manual lock. A member that was locked using the **L** line command in Compuware/VF or by using the LOCK batch command. Members that have been manually locked retain their locks during the autodeletion process. See also **automatic lock**.

MIPS. Millions of Instructions per Second. A unit of measure of processing performance equal to one million instructions per second.

MSU. Millions of Service Units per Hour.

Oldest member. (1) For report files, the diagnostic report with the earliest abend date and time. (2) For source listing files, the program with the earliest compile date and time.

PL/I Language Processor. One of four language processors provided by Compuware. This language processor accepts PL/I compiler output, builds sort work records and an incore symbol table of all of the identifiers, and produces Processor control blocks that can then be used as input to other Compuware products.

Primary Command. A command entered in the COMMAND INPUT field.

Problem Documentation Utility. A CSS utility that captures the documentation needed by Compuware for resolving technical issues and problems. This utility allows you to collect the documentation — such as copybooks, SYSPRINT, and DDIO members — that CSS Technical Support requires for problem resolution.

Report File. A file containing diagnostic reports that is accessed by DDIO.

Security Exit Program. A user-coded program that allows or denies execution of a requested command for a selected member by a specific user against particular DDIO files.

Server Grace. The amount of time an LPAR can run without being connected to the server.

Shared Directory. A variable-length record VSAM RRDS that maintains information about abends and language processing along with the attached database activity. A shared directory can contain Abend-AID for CICS directory records for each region and transaction dump known to a server, Abend-AID directory records for abend report processing, or source listing shared directory records necessary to process source listing database members.

SMP/E. System Modification Program/Extended is IBM's standard facility for installing and maintaining software modifications in the MVS environment.

Source Listing. A compiled listing and other information about a program stored in a source listing file.

Source Listing Database. A specially formatted source listing file, owned and managed by a shared directory, that is used by the C Language Processor for C LONGNAME support.

Source Listing File. A file containing source listings that is accessed by DDIO.

STROBE. The STROBE MVS Application Performance Measurement System is a product that determines where and how application time is spent in online regions and batch processing programs and how system resources are used. STROBE collects several types of data as it tracks activity within an MVS environment and produces a collection of reports that helps you determine where to revise applications to improve their performance.

Suppressed statements. Statements that are not displayed in the compiler listing. Suppressed statements in a DDIO source member contain an internal flag that the XPEDITER/TSO product can use to display or suppress source statements using the GEN command.

Transaction Database. A DDIO file, containing transaction reports, that is managed by a shared directory in Abend-AID for CICS. Multiple transaction reports are stored in a single transaction database.

Transaction Report. A Abend-AID for CICS report created from CICS transaction terminations that is stored in a DDIO file.

Unlocked Member. A DDIO file member that is not currently locked. The locked member can be unlocked either manually, using the batch file utility or Compuware/VF, or automatically as a result of AUTODELETE=DUPS or AUTODELETE=STAGED processing. An unlocked member in a full DDIO file formatted with AUTODELETE=YES, AUTODELETE=DUPS, or AUTODELETE=STAGED can be automatically deleted when another member is added to the file.

Verification Report (LMVERIFY). Displays the result of a verification program that checks out each product version in cache. This data is exactly what a product would receive when it runs. Running this report will prove that the License Management System environment is properly set up for the customer.

View time. The time when an diagnostic report or source listing is presented in a readable format for viewing online or printing.

Index

Special Characters

A

B

C

D

E

F

G

H

I

J

L

M

N

O

P

R

S

T

U

V

W

X